

64'er

JUNI 1984

ÖS 50,- /Sfr 6,- DM 6,-

684 DAS MAGAZIN FÜR COMPUTER-FANS

Super-Basic, starke Farbgrafik

Test: C 16

Der VC 20-Nachfolger?

Die besten Textprogramme

4 Renner im Test
Marktübersicht
Superprogramm zum Eintippen

Daten auf Band

Es muß nicht immer Datasette sein

Tips, Tricks und billige Alternativen

Drucker für Anspruchsvolle

Der Epson am C 64

Listing des Monats:

Mit dem VC 20 in den Untergrund

Die Exklusiv-Story

Kein Spiel ist sicher vor dem Knacker-Club



Programmiervettbewerb Utilities.
Der Trick mit dem Autostart.
Jede Menge Listings, Tips
und 3 interessante
Kurse



synapse

SOFTWARE

ariolasoft

Steinhauser Straße 3
8000 München 80



ZAXXON
Diskette und Cassette für
Commodore 64



DIMENSION X
Diskette und Cassette für
Atari 400/600XL/800/800XL



BLUE MAX
Diskette und Cassette für
Commodore 64
Cassette für
Atari 400/600XL/800/800XL



RAINBOW WALKER
Diskette und Cassette für
Atari 400/600XL/800/800XL

Außerdem sind lieferbar:

PHARAOH'S CURSE

Diskette und Cassette für Commodore 64
Diskette und Cassette für
Atari 400/600XL/800/800XL

NECROMANCER

Diskette und Cassette für
Atari 400/600XL/800/800XL

PROTECTOR II

Diskette und Cassette für Commodore 64

DRELBS

Diskette und Cassette für Commodore 64

SENTINEL

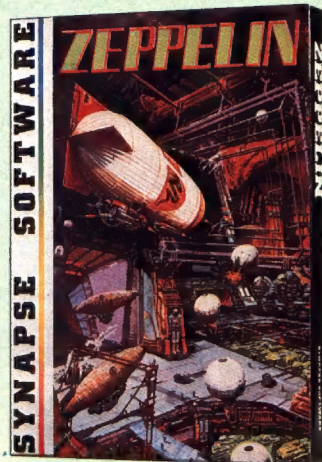
Diskette und Cassette für Commodore 64

SURVIVOR

Diskette und Cassette für Commodore 64

SHAMUS

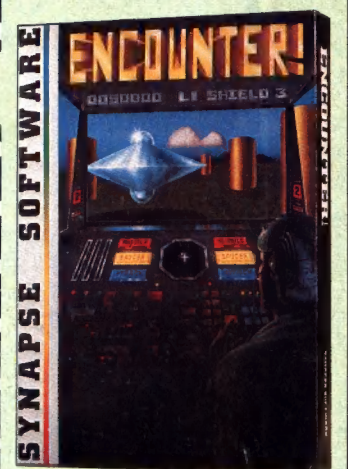
Diskette und Cassette für Commodore 64
Steckmodul für Atari 400/600XL/800/800XL



ZEPPELIN
Diskette und Cassette für
Atari 400/600XL/800/800XL
Diskette für Commodore 64



FORT APOCALYPSE
Diskette und Cassette für
Commodore 64
Steckmodul für
Atari 400/600XL/800/800XL



ENCOUNTER
Diskette und Cassette für
Atari 400/600XL/800/800XL

... eigentlich schon Spiele von morgen

Aktuell

Die Neuigkeiten von der Hannover-Messe 84 8

Die Exklusivstory

Kein Spiel ist sicher vor dem Knacker-Club

Der Sumpf: Wo statt »Copy-right« beispielsweise »Broken by Antiram« steht, da waren die Knacker am Werk 12
Kleinanzeigen mit Zündstoff 14

Test

Super-Basic, starke Farbgrafik

Test: C 16

Der VC 20-Nachfolger? 20

Hardware

Kopplung zwischen zwei VC 20: Experimente mit der Datenübertragung 24

Daten auf Band. Es muß nicht immer Datasette sein:

Tips, Tricks und billige Alternativen 30

Drucker für Anspruchsvolle:

Der Epson am C 64 34

Software

Strukturiertes Programmieren (3) 37

Software-Test

Eine tolle Sache: Exbasic Level II: 42

Die besten Textprogramme

Marktübersicht: Textverarbeitungsprogramme 47

Test: SM-Text 64 48

Test: Wordpro 3 Plus 52

Test: Blitztext 54

Spiele-Test

AE — ein Spiel mit Pfiff 56

Dino-Eggs 57

Programme zum Abtippen

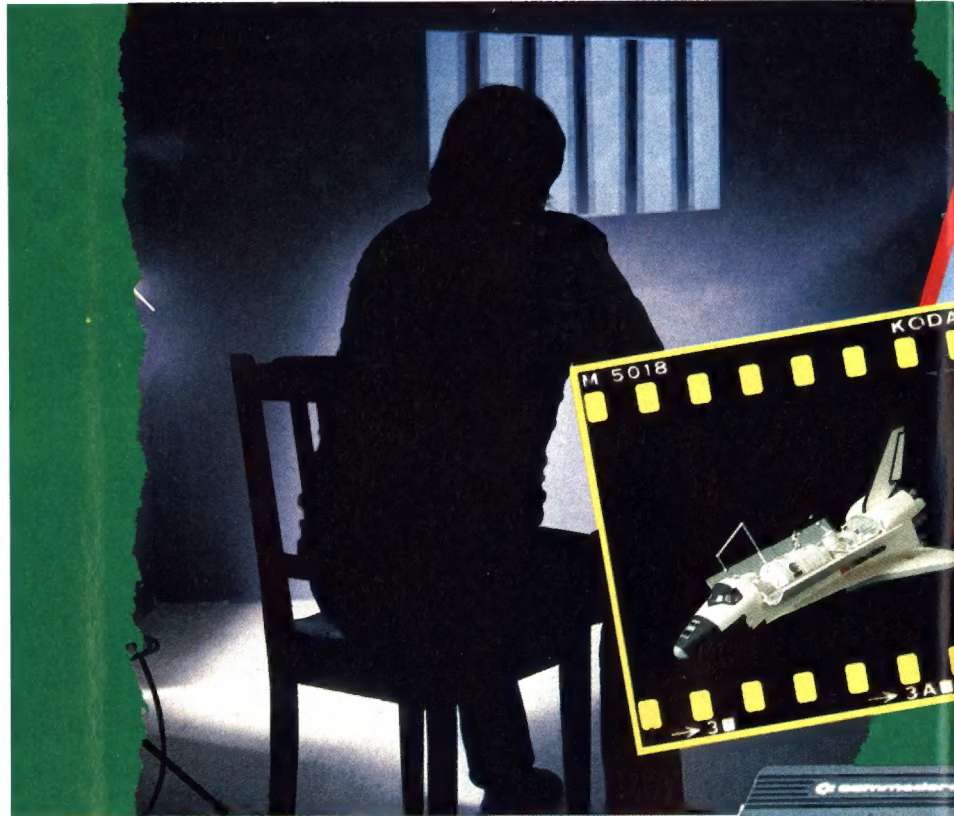
Anwendungen

Quicktext ein Textprogramm für C 64 60

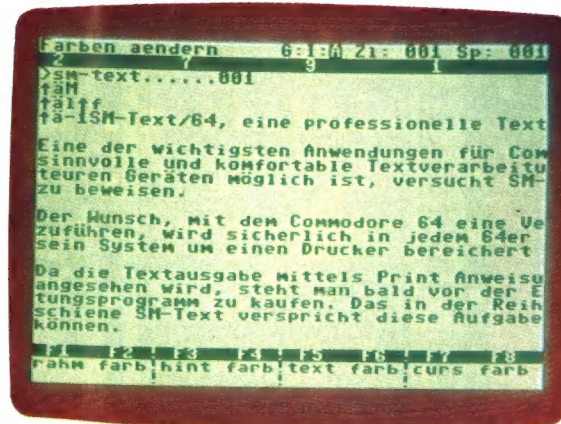
Die Anwendung des Monats: Lehrerkalender (C 64) 64

Supervoc: Ein Programm, mit dem man leicht Vokabeln lernen kann (C 64) 69

Kurz-lang-kurz — kein Problem: Morsetrainer (C 64) 72



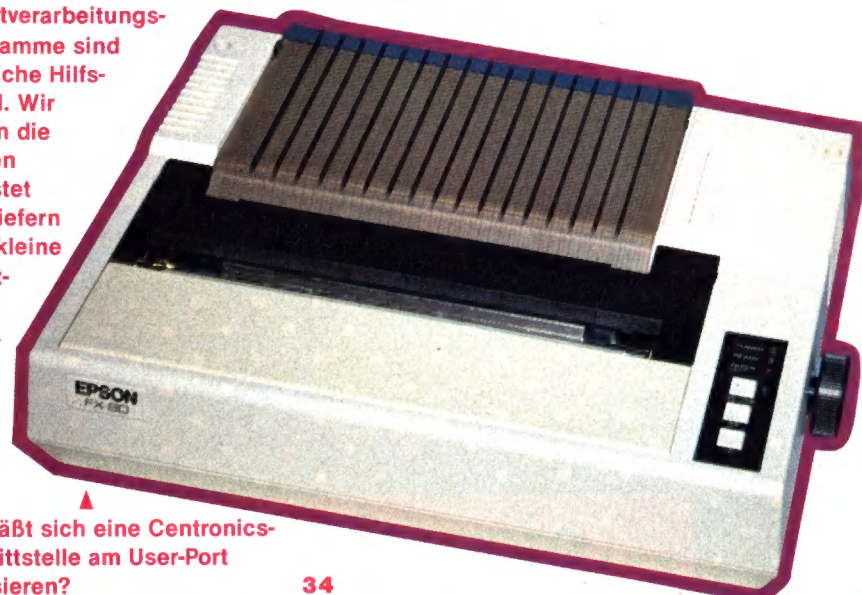
▲ Exklusivinterview: Vor den Knackern ist kein Programm sicher 12



▲ Textverarbeitungsprogramme sind nützliche Hilfsmittel. Wir haben die besten getestet und liefern eine kleine Marktübersicht. 47



Der neue C 16 im alten Gewand



▲ Wie läßt sich eine Centronics-Schnittstelle am User-Port realisieren? 34

▼ Es muß nicht immer Datasette sein!

30



◀ So schießen
Computer-Besitzer
bessere Fotos 138



and. Ist er der Nachfolger vom VC 20? 20



▲ Das Listing des Monats ist ein faszinierendes Spiel für den VC 20. Begeben Sie sich in den Untergrund von New York.

120

Grafik

Movemaster: der VC 20 als Trickfilmstudio	78
Grafische Darstellung statt Zahlenfriedhof (C 64)	82

Spiele

Hot Wheels (C 64)	98
Ghost Manor (VC 20)	104

Tips & Tricks

Steckmodule auf Kassette (VC 20)	107
Logic Disassembler (VC 20/64)	108
Synthetische Steuerzeichen (C 64)	114
RESET für den VC 20	116
Der Trick mit dem Autostart (C 64)	117
Tips und Tricks für den VC 20	118
Veränderungen am Basic (C 64)	119

Listing des Monats

Mit dem VC 20

in den Untergrund

Eine Kombination aus Action- und Abenteuerspiel	120
---	-----

Kurse

Was nicht im Handbuch steht, Kurse zum Mitmachen

Strubs — ein Precompiler für Basic-Programme (3)	128
Reise durch die Wunderwelt der Grafik (3)	132

So machen's andere

Der VC 20 als Fotolehrling	138
----------------------------	-----

Wettbewerbe

Programmierungswettbewerb 2000 Mark Preise für das Erstellen einer Programm-Bibliothek	144
Superchance: 2000 Mark für das Listing des Monats	146
Wir suchen die Anwendung des Monats: 500 Mark zu gewinnen	147

Rubriken

Editorial	8
Leserforum	18
Druckfehlerteufelchen	124
Bücher	127
Vorschau	151



Hick-Hack

Da gibt es ein Kopierprogramm, mit dem man die 1541-Floppy in knapp 5 Minuten formatieren und kopieren kann. Das geht also wesentlich schneller als nach dem Original-Commodore-Modus und bietet für viele die ersehnte Arbeitserleichterung. Leider kann man damit nicht nur die eigene Arbeit rationalisieren, sondern auch fremde Programme kopieren. Vorstellen oder tuschweigen? Man kann in einer Schlosserzeitung nicht das Thema Schloß und Schlüssel mit der Begründung ignorieren, die Zeitung könne ja auch von Einbrechern gelesen werden. Andererseits wird eindeutig zuviel kopiert, was nicht kopiert werden dürfte — da müssen die Hilfsmittel nicht unbedingt in allen Details beschrieben werden. Für viele Köpfer stellt allerdings das Knacken eines List- oder Kopierschutzes eine denksportartige Herausforderung dar — unehrliche Ziele werden nicht verfolgt.

Sollte man also Kopierv Verfahren und Knacktechniken nicht lieber offenlegen — wenn man sie schon nicht mit Simsalabim verschwinden lassen kann? Am besten kann sich ja jeder vor Risiken schützen, wenn er die Gefahren genau kennt (und gegebenenfalls die Strafen). Die Mehrzahl der in Deutschland zugelassenen Autos dürfte auf Spitzengeschwindigkeiten über 140 km/h kommen. Man darf sie ohne weiteres auch in der Stadt oder auf der Landstraße benutzen — vom Autofahrer wird ganz selbstverständlich verlangt, daß er sich an die jeweils geltenden Geschwindigkeitsbegrenzungen hält. Warum sollte man vom Computerbenutzer soviel weniger verlangen können als vom Autofahrer? Auch wenn es noch keinen Computerführerschein und kein Byte-Radar gibt.

M. Pauly, Chefredakteur

Aktuell Die Neuigkeiten

von der

Zwei Unternehmen haben auf der Hannover-Messe '84 für Furore gesorgt.

Zum einen war dies Apple mit dem Macintosh und zum anderen natürlich

Commodore mit den neuen Heimcomputern C 16 und C 264, sowie einigen

Systemen der gehobenen Preisklasse.



Bild 1. Der C 264 mit neuem Joystick

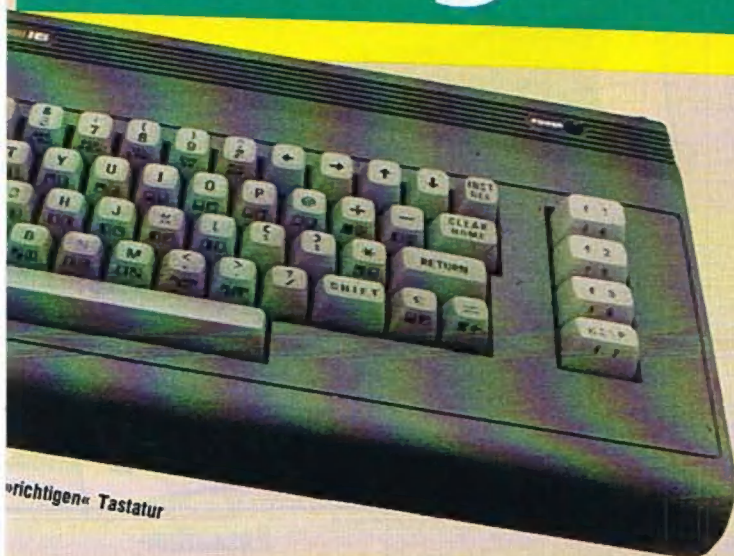
An erster Stelle sind selbstverständlich die beiden neuen Heimcomputer C 16 und C 264 zu nennen. Offiziell wurden sie erst auf der Messe vorgestellt. Die 64'er Redaktion hat aber den C 264 bereits in der letzten Ausgabe getestet (Bild 1), der C 16 (Bild 2) beziehungsweise C

116 wird in dieser Ausgabe behandelt. Dieser C 16 (die Version C 116 mit der Gummitastatur wird in Deutschland voraussichtlich nicht angeboten werden) weist ein äußerst günstiges Preis/Leistungsverhältnis auf, er wird etwa 350 Mark kosten. Die Zielrichtung des C 16 ist laut



Bild 2. Der C 16 kommt in Deutschland mit e

Hannover Messe' 84



«richtigen» Tastatur

Alwin Stumpf, Geschäftsführer von Commodore Deutschland, den Computer-Anwender von morgen schon heute mit Computern vertraut zu machen. Der C 16, ausgestattet mit 16 KByte RAM und 32 KByte ROM und der erweiterten Basic-Version 3.5, einer »physikalischen« Cursorsteuerung, einem Format das kleiner als A4 ist, einer Schreibmaschinen-Tastatur, mit Umstellungsmöglichkeit von 40 Zeichen x 25 Zeilen und 121 Farbtönen, einem Synthesizer mit jeweils einem Ton- und Rauschgenerator und der Aus/Eingabe über serielle Schnittstelle, zwei Joystickports (allerdings mit neuartigen Anschlüssen), Video- und TV-Anschluß so-

wie dem Kassettenport, soll eine sinnvolle Abrundung der Commodore-Palette nach unten realisieren. Inwieweit der C 16 den VC 20 ersetzen oder ablösen wird ist fraglich, zumindest jedoch zu vermuten.

C 16 und C 264 — die neuen Renner?

Praktisch als Gegenpol soll der neue C 264 laut Harald Speyer, Vice President des Unternehmens, beruflichen Zwecken dienen und den Markt nach oben hin abdecken. Zu diesem Bereich gehören zum Beispiel Dateiverwaltung, Textverarbei-

tung, Tabellenkalkulation und Lösungen zu Steuer-, Finanzierungs- und Versicherungsfragen, sowie Grafik- und Window-Fähigkeit und zwar zu Hause wie im Büro. All diese Funktionen soll der C 264 mit Hilfe des Softwarepakets »Magic Desk« — als Modul oder Diskette ladefähig — bewältigen können. Dieses und weitere Programme sollen künftig wahlweise auch gleich in der Hardware eingebaut be-

Tongeneratoren und einem Rauschgenerator, die Audio-schnittstelle, Sprites und der Maschinensprach-Monitor können Sie im Testbericht nachlesen. Die gesamte Peripherie vom C 64 wird auch mit dem C 264 nutzbar sein. Commodore sieht den C 264 nicht als Konkurrenz für den C 64, sondern als Ergänzung nach oben hin. Der Preis wird sich demzufolge auch etwas noch oben verschieben und wohl in der Gegend



Bild 3. Mit dem MCS 801 farbig drucken

zogen werden können (sogenannte integrierte Software). Die Unterschiede zum C 64 und C 16 wie etwa 64 KByte RAM (davon 60 KByte für Basic), Synthesizer mit zwei

von 1200 Mark zu liegen kommen.

Nun gibt es auch von Commodore einen Farbdrucker (Bild 3). Dieser Color-Matrixdrucker MCS 801 ver-

fügt über folgende Charakteristika:

7 Farben, serielles Commodore-Interface, 50 Zeichen pro Sekunde, mit maximal vier Floppy-Laufwerken funktionsfähig, Groß/Kleinschreibung sowie Grafikzeichen. Der Preis wird sich wohl bei zirka 1500 Mark einpendeln.

Drei neue Drucker

Der Typenradrunder DPS 1101 (DPS steht für Daisy Wheel Printer System; Bild 4) entspricht hinsichtlich Bedienerfreundlichkeit,

Druckqualität und des Verarbeitungskomforts dem Stand der Technik. Er bietet bidirektionale Druckweise mit Triumph Adler-kompatiblen Typenrädern, druckt bis zu 18 Zeichen in der Sekunde, und beherrscht Proportionalsschrift; die maximale Breite inklusive Traktorführung beträgt 13 Zoll, es lassen sich bis zu zwei Kopien erzeugen und er ist sowohl an VC 20, C 64 und C 264 anschließbar. Mit rund 2000 Mark liegt er noch am unteren Rand der Preisskala.

Außerdem werden demnächst noch zwei neue Matrixdrucker, der MPS 802 und 803 (Bilder 5 und 6) auf den Markt kommen. Über diese beiden Drucker war aller-

dings auf der Hannover Messe noch nicht viel zu erfahren.

Daß durch ein Modul im Wert von 250 Mark der Commodore 64 (250 000 Exemplare sind mittlerweile in der Bundesrepublik im Einsatz) Btx-fähig wird, sofern ein Farbfernsehgerät mit einem CEPT-Dekoder vorhanden ist (Bild 7, mit einem Loewe-Gerät), haben wir schon in der letzten Ausgabe beschrieben. Die Vorteile dieser Lösung sind die Speicherung der Btx-Seiten auf Floppy und die Off-line-Benutzung des Btx-Netzes. Das heißt, sie können die Antwortseiten ohne Zeitdruck ausfüllen und so die üblichen Fehlerraten reduzieren.

Sicher werden sich einige gewundert haben, warum

bei dem Programm Magic Desk die einzelnen Symbole für Schreibmaschine, Drucker oder Papierkorb so groß ausgefallen sind. Die Erklärung ist, daß nun auch ein Touch-Panel (Bild 8) auf den Farbmonitor 1701/1702 aufgesteckt werden kann. Dadurch lassen sich nun die einzelnen Funktionen durch einfaches Zeigen mit dem Finger aufrufen.

Speziell für den Bereich Schule, Aus- und Fortbildung ergab sich durch die Zusammenarbeit von Commodore mit dem Schulbuch-Verlag Westermann eine ganze Menge an neuen Lernprogrammen. So gibt es jetzt für den C 64 einen Rechtschreiblöwen, einen Rechenlöwen (mit den vier Grundrechenarten) für das zweite und dritte Schuljahr, ein Mathematikprogramm für das erste Schuljahr, die leicht zu erlernende Pro-



Bild 5. Ein neuer Matrixdrucker 802. Daten unbekannt



Bild 6. Der Matrixdrucker MPS 803



Bild 8. Durch das Touch-Panel wird der Monitor endlich »berührbar«

Kassetten-Betriebssystem für Commodore 64

Unter der Bezeichnung ACOS+ (Advanced Cassette Operating System) bietet die Firma Melbourne House ein umfangreiches Utility-Paket für den C 64 an. Kernstück von ACOS+ ist ein neuartiges Betriebssystem für die Datassette, das alle Kassettenoperationen wesentlich beschleunigt. Ähnlich wie bei einer Diskette wird ein Directory am Anfang jeder Kassette aufgezeichnet. Nach Auswahl eines Programms spult ACOS+ automatisch mittels schnellen Vorlaufs bis zur entsprechenden Bandstelle vor und lädt das gewünschte Programm. Daneben enthält ACOS+ zahlreiche weitere Hilfsmittel für Spritesteuerung, Tonerzeugung und Grafik sowie eine Reihe weiterer Utilities.

Info: Melbourne House (Publishers) Ltd., Castle Yard House, Castle Yard, Richmond TW10 6TF (USA), Tel. 01-940-6064.

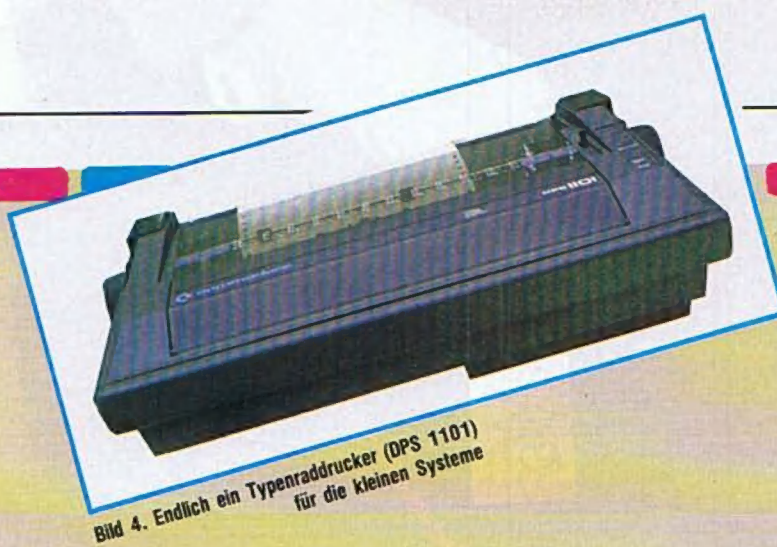


Bild 4. Endlich ein Typendrucker (DPS 1101) für die kleinen Systeme

grammiersprache Logo und als Ergänzung dazu Logo-Grafik mit dem durch einfache Befehle komplexe Zeichnungen erstellt werden können. Durch diese Initiative soll der doch erhebliche Nachholbedarf im computergestützten Lernen in Deutschland überwunden werden. Dazu sollen auch eine Reihe von Lehr- und Arbeitsbüchern beitragen, so zum Beispiel »Logo« (Band 3, aus der neuen Commodore-Buchreihe), ein Lexikon und »Werkzeug« für den Lernenden im Umgang mit der Programmiersprache Logo. Natürlich hatte Commodore nicht nur Neuigkeiten auf

dem Heimcomputer-Sektor anzubieten. Besondere Beachtung fand auch der IBM-kompatible Commodore PC. Wer darüber genauere Informationen haben will, kann einen ersten Testbericht in Computer persönlich, Ausgabe 11, lesen. Endgültig zu haben wird der tragbare 16-Bit-Computer wohl aber erst im Herbst sein.

Speziell für den kommerziellen Markt kreierte Commodore den CBM 8296, eine Weiterentwicklung des altbewährten CBM 8032/8096. Er ist ausgestattet mit 128 KByte RAM; das Betriebssystem

mit acht und mehr Terminals, ermöglichen. Der Z 8000 stellt also ein gehobenes

von 1024 x 1024 Punkten, zwei programmierbare RS 232-Schnittstellen, selbstverständlich das IEEE/84-Interface, eine optionale 10-MByte-Festplatte, 1,3 MByte-Floppy, arithmetischer Koprozessor Z 8070 und Maus-Anschluß.

Commodore forever?

Commodore hat also auf der Messe voll zugeschla-



Bild 7. Der Commodore 64 kann als intelligente Tastatur für die Bearbeitung von Btx-Seiten eingesetzt werden. Die Abspeicherung auf die Floppy ist dabei möglich

Neuer 6502-Editor/Assembler

Von Profisoft, gibt es jetzt ein neues Editor/Assembler-Paket speziell für den C 64. Das Programmpaket enthält einen komfortablen 2-Pass-Assembler, einen erweiterten Editor, einen Disassembler sowie einen Maschinensprachemonitor für direkten Speicherzugriff. Der Assembler unterstützt den vollen 6502/6510-Befehlssatz. Der sehr komfortable Editor bietet eine Vielzahl von Funktionen zur Bearbeitung von Assembler-Quellprogrammen, kann aber auch für Basic-Programme eingesetzt werden. Alle Editorfunktionen sind direkt von Basic aus erreichbar. Der Disassembler ermöglicht die Definition eigener Label und erzeugt assemblierbaren Quellcode. Der Editor/Assembler wird wahlweise auf Kassette oder Diskette geliefert.

Info: Profisoft, Sutthausen Str. 80/82, 4500 Osnabrück, Tel. (0541) 53905.

stem LOS-96 gehört zum Lieferstandard, Programmiersprachen sind TCL-Pascal, UCSD-Pascal, Comal und Assembler, — wählbar sind ASCII- oder DIN-Tastatur, die Tastatur ist frei beweglich.

Mit dem CBM Z 8000 präsentiert Commodore eine neue Computerfamilie die in der Standardversion den Anschluß von zwei Arbeitsplätzen gestattet; spätere Ausbaustufen sollen die Verbin-

16-Bit-Multi-User-System das unter dem UNIX 7 angepaßten Coherent-Betriebssystem COHOS läuft, dar. Weitere Kennzeichen sind 256 KByte RAM minimale Speicherkapazität, Programmiersprachen wie Basic, Pilot, Assembler und C-Compiler optional, ein monochromer Bildschirm mit einer Auflösung

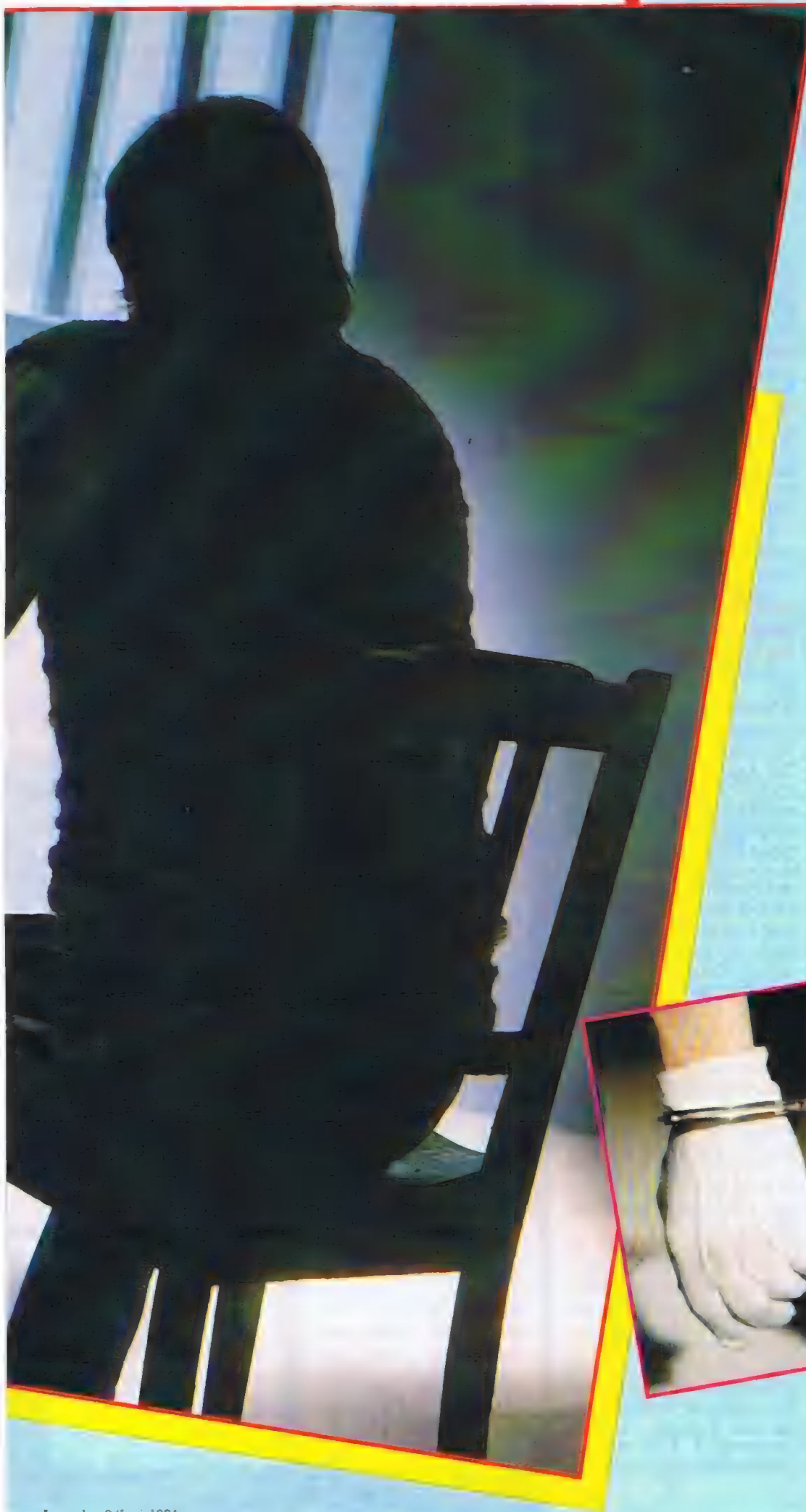
gen. Nach meiner persönlichen Beurteilung war kein anderer Stand derart überlaufen, erregte kein anderes Unternehmen dieses Aufsehen. Mit Commodore wird also auch in der Zukunft zu rechnen sein. (aa)

Nach eigenem Empfinden
machen sich die Raubko-
piierer die Hände nicht
schmutzig

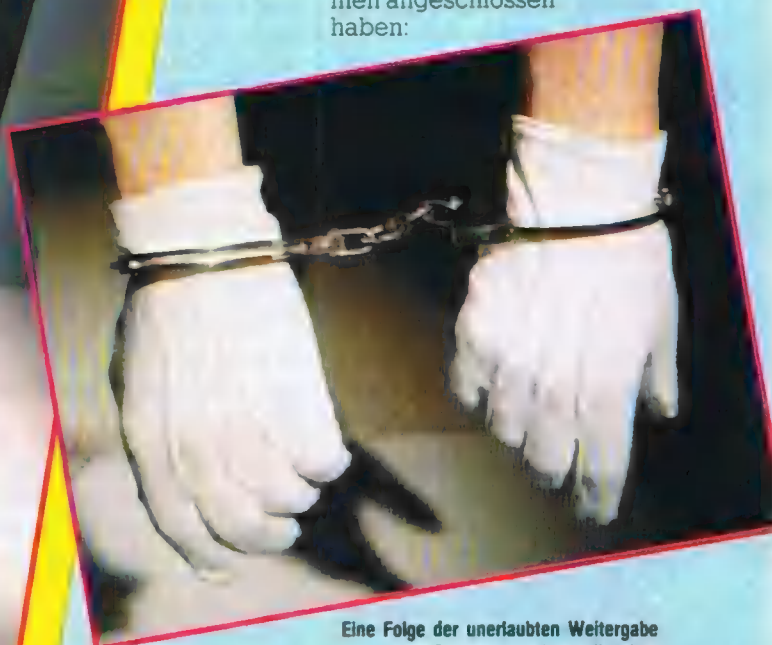


Der Sumppf

Wer hat nicht schon selbst
Programme zumindest gesehen,
in denen die Kürzel
Jedi, KBR, Kotzbrocken, Oleander,
oder 1103, oder Nachrichten
wie »Brooken by Antiram.
Greetings to Zeropage
and Superman« von der dunklen
Herkunft zeugen?
Wir haben uns in den Untergrund
begeben und mit der
»High Society« der deutschen Knacker
gesprochen. Die Creme de la creme der Knacker, rund ein
Dutzend Leute, haben sich zu einem »Club«
zusammengeschlossen.



Das wohl überraschendste Ergebnis dieses Interviews dürfte die Aussage der Knacker sein, künftig mit einigen Händlern zusammenarbeiten zu wollen. Wie kann es dazu kommen, daß die Geschädigten mit den Schädigern kooperieren? Ein derartiges »Gentleman Agreement« wird nur geschlossen werden, wenn beide Seiten einen gewissen Vorteil daraus ziehen können. Der Vorteil für die Knacker ist offensichtlich. Sie werden direkt von den Händlern mit der neuesten Software versorgt. Die Knacker versprechen im Gegenzug diese Programme zumindest eine Zeitlang nicht weiter zu verbreiten, oder nur im engsten Kreis. Somit hat der Händler eine ausreichende Zeitspanne zur Verfügung, das Programm in kostendeckenden Mengen zu verkaufen — und das geht nur, wenn es noch nicht auf dem Schwarzmarkt zu haben ist. Außerdem beraten ihn die Knacker, welche Programme er sich aus den USA oder England besorgen soll. Die Knacker haben ja die meiste Erfahrung und oft auch die besseren Beziehungen und wissen deshalb auch, welche Programme sich gut verkaufen lassen und welche nicht. Soweit so gut, aber es gibt auch noch eine andere Sorte, die sich nicht diesem Abkommen angeschlossen haben:



Eine Folge der unerlaubten Weitergabe von Programmen kann die oben dargestellte sein



die Raubkopierer. Um den Unterschied aufzuzeigen soll eine Definition der Knacker verwendet werden. Originalton:

»Knacker oder Cracker sind Leute, die den Kopierschutz entfernen, so daß man das Programm einfach mit LOAD und SAVE vervielfältigen kann. Zudem werden die Programme »komprimiert«, das heißt die platzaufwendige Abspeicherung des Originals wird reduziert, damit aus Gründen der Übersichtlichkeit mehr Spiele auf eine Diskette passen (kein Wunder bei mehr als tausend Programmen).

Kopierer hingegen haben spezielle Kopierprogramme (Block-Copy, F-Copy), mit denen der Kopierschutz einfach mitübertragen wird. Die wollen durch den Verkauf dieser Programme Geld machen.

Und dann gibt es in Deutschland seit neuestem die Hacker. Diese dringen über Modem und Akustikkoppler in Datenbanken anderer Leute ein und schauen sich da ein bißchen um. Bei Eduscho werden sich, so haben wir gehört, in nächster Zeit einige Leute über eine unerwartete Gehaltserhöhung freuen können. Es gibt mittlerweile sogar ein eigenes Magazin, die Datenschleuder vom Chaos Computer Club in Hamburg. Da stehen prima Tips und Telefonnummern drin.

Nach dieser kurzen Darstellung, wer eigentlich was macht, folgt nun ein Auszug aus dem Interview mit vier Meister-Knackern. Sie haben sehr freimütig aus ihrem Metier erzählt. Aus verständlichen Gründen können wir keine Namen nennen.

Frage: Wie sind diese Kopierer eigentlich zu den Kopierprogrammen gekommen?

»Tja, die Kopierprogramme stammen von Leuten, die Maschine können (das heißt, sehr gut in Maschinenspra-

che programmieren), aber über keine so guten Beziehungen zu den Händlern verfügen. Die haben sie dann an Leute gegeben, die gute Beziehungen zu den Firmen, aber nicht die Fähigkeiten hatten. Die haben dann zwar die begehrten Spiele, aber leider sind so die Kopierprogramme auf den Markt, ich meine natürlich den Schwarzmarkt, gekommen. Früher war der Schwarzmarkt noch von den Knackern abhängig, doch jetzt sind leider die Kopierprogramme verbreitet.«

Wenn bereits Programme im Umlauf sind die den Kopierschutz einfach mitkopieren, hat es da überhaupt noch Sinn, Programme zu schützen?

»In den USA werden die besten Knacker von den Software-Firmen aufgekauft, um einen Kopierschutz für deren Produkte zu schreiben. Die Knacker haben ja das Know-how. Wir haben auch schon einen Kopierschutz gemacht, der nicht mit den gängigen Kopierprogrammen zu knacken ist. Wir bieten daher den Firmen unsere Mitarbeit an. Der Vorteil für die Firmen liegt klar auf der Hand. Nur noch die besseren Programmierer können die neuesten Programme knacken und die wollen, zumindest was uns betrifft, mit den Händlern zusammenarbeiten. Ein guter Kopierschutz hat aber wenig Sinn, wenn andere Leute die Programme schon ungeschützt verbreiten. Wir hatten zum Beispiel die ungeschützte Version von Soccer von Commodore bereits ein halbes Jahr bevor es auf den Markt kam.«

Vertreibt ihr noch geknackte Programme?

»Früher haben wir schon einige Sachen verkauft, jetzt aber nicht mehr so. Ein bißchen schon noch, wir haben zum Beispiel eine ziemlich hohe Telefonrechnung durch die ständigen Verbindungen mit den Freunden in ganz Deutschland. Der Informationsfluß zwischen uns ist sehr

gut. Sobald einer ein neues Spiel hat, ist das innerhalb von einem Tag rum. Die Unkosten müssen ja noch reinkommen. Aber ansonsten tauschen wir nur im harten Kern. Profit wollen wir keinen machen.«

Warum knackt ihr eigentlich Programme?

»Zum einen weil es einfach Spaß macht in fremden Programmen rumzupfuschen, und zum anderen weil die Programme einfach viel zu teuer sind. Man kann nicht 100 bis 150 Mark für ein Spielprogramm ausgeben und dann nach zwei oder drei Tagen ist es langweilig. Wir sind Sammler, wir müssen haben. Je mehr desto besser und da kann man sich das natürlich als Schüler oder Student nicht leisten. Also wird ein Spiel gekauft oder besorgt, geknackt und getauscht und schon haben sich die Kosten vermindert.«

Wie ist eigentlich das Verhältnis von Spielprogrammen und ernsthaften Programmen in eurer Sammlung?

»Wir haben hauptsächlich Spiele. Ernsthafte Programme haben wir auch ein paar, aber nur so zum Hausgebrauch. Diese Programme sind auch oft zu komplex um sie ohne Anleitung sinnvoll einsetzen zu können. Wir sind Spieler. Am liebsten sind mir die Denkspiele und die Abenteuerspiele. Schießspiele werden schnell langweilig.«

Welches sind die wichtigsten Werkzeuge zum Knacken?

»Natürlich ein bis zwei Computer, Monitore, Floppy-Laufwerke, und einige Toolkits wie Maschinensprach-Monitore und ein guter Disk-Mon. Am wichtigsten ist aber ein großes Hirn.«

Habt ihr eigentlich keine Gewissensbisse?

»Nein, keine, wieso?«

Einige Firmen gehen mittlerweile rigoros gegen die Raubkopierer und

Kleinanzeigen

Wer kennt

sie nicht, die Kleinanzeigen

mit den verlockenden

Angeboten: Über 300 Programme für nur

100 Mark. Daß

hier manches nicht mit rechten

Dingen zugeht,

sieht auch der Laie — wenn zum

Beispiel dabeisteht »alle Synapse

Eine Programmsammlung von 300 Programmen dürfte in der Regel durch Softwaretausch entstanden sein. Kann man aber die derart erworbenen Programme weiterverkaufen? Die Antwort lautet ganz klar: nein. Das Copyright und die Vertriebsrechte liegen bei der Herstellerfirma oder dem

Vertreiber. Dennoch wird aus Sammlerleidenschaft oder wegen niederer Profitier fleißig kopiert und verkauft. Gegen private Kopier- und Tauschaktionen im Freundeskreis können die Firmen wegen der Unübersichtlichkeit des Marktes nur wenig unternehmen. Aber gegen Gewerbetreibende sind



Fragen Sie doch!

Selbst bei sorgfältiger Lektüre von Handbüchern und Programmbeschreibungen bleiben beim Anwender immer wieder Fragen offen. Viel mehr Fragen ergeben sich bei Computer-Interessenten, die noch keine festen Kontakte zu Händlern, Herstellern oder Computerclubs haben. Sie können der Redaktion Ihre Fragen schreiben oder Probleme schildern (am einfachsten auf der beigehefteten Karte). Wir veranlassen, daß die Fragen von einem Fachmann beantwortet werden. Allgemein interessierende Fragen und Antworten werden veröffentlicht.

Komma als Satzzeichen

Frage: „Kann man bei einem Inputbefehl das Komma als normales Satzzeichen verwenden?“

Ausgabe: 64/5

Name: Gerhard Giessmann

Um einen INPUT-Befehl auch Kommata (und Doppelpunkte) akzeptieren zu lassen, muß man lediglich als erstes Zeichen ein Anführungszeichen eingeben, das nicht in die Variable nach »INPUT« übernommen wird. Diese Eingabe läßt sich auch automatisch über den Tastaturpuffer simulieren, indem man »POKE 631, 34: POKE 198, 1« vor den »INPUT« setzt. Jetzt wird das Anführungszeichen von selbst ausgegeben. Eleganter ist es natürlich, wenn eine eigene Eingaberoutine verwendet wird, wie beispielsweise »INPUTFORM« und »INPUTLINE« im EXBASIC LEVEL II.

Ralph Babel

VC 20-Programme auf CBM 8032

Frage: „Wie kann ich Programme von VC 20 auf dem CBM 8032 zum laufen bringen?“

Ausgabe: 64/5

Name: Gerhard Grahl

Das Einlesen von Relativprogrammen (abgespeichert ohne Sekundäradresse) auf Kassette mit beliebiger Startadresse in den CBM 8032 erfolgt durch OPEN 1: POKE 635, 1: POKE 636, 4: SYS 62456

Das Programm kann nun gelistet werden (allerdings beherrschen die Drucker zum 80er meist nicht das Pfundzeichen, was bei Ausgabe von »Control Red« beachtet werden muß). Mit Exbasic Level II können Diskettenprogramme durch »MERGE« in den 80er geladen werden.

Ralph Babel

Nach einer Stunde Fernseher justieren

Frage: „Nach zirka einer Stunde verliert mein Fernseher die Farbe, was kann ich dagegen tun?“

Ausgabe: 64/5

Name: Klaus Heinz

Die einfachste Lösung dieses allgemein verbreiteten Problems erscheint mir im Nachjustieren der Sendereinstellung am Fernseher (was dann von Zeit zu Zeit mal geschehen muß).

Ralph Babel

Das Problem kann auch an einem verstellten Trimmer des C 64 liegen. Dieser befindet sich auf der Rückseite des Computers und kann mittels eines kleinen Schraubendrehers von außen verstellt werden.

Guido Stapel

Rechengenauigkeit

Wieso kommt beim Arbeiten mit der Sinus-Funktion beim VC 20 ein anderer Wert heraus als mit meinem Taschenrechner?

Albert Bartels

Die meisten Basic-Interpreter rechnen intern mit einer geringeren Genauigkeit als ein Taschenrechner. Besonders bei höheren mathematischen Funktionen kann es daher zu kleinen Abweichungen kommen.

Commodore

64-Programme auf VC 20?

Sind die für den C 64 abgedruckten Programme auch für den VC 20 zu benutzen? Natürlich nur, soweit die benötigte Peripherie vorhanden ist.

Andreas Volz

Sofern es sich um reine Basic-Programme ohne POKES und PEEKs handelt, gibt es keine Schwierigkeiten bei der Übernahme von Programmen des C 64 auf den VC 20 oder umgekehrt. Wegen der unterschiedlichen Zeichenzahl pro Bildschirmzeile (22 Zeichen beim VC 20, 40 Zeichen beim C 64) wird die Bildschirmdarstellung unter Umständen jedoch etwas unübersichtlich. Entsprechende Änderungen bei den PRINT-Befehlen sind jedoch auch für den Anfänger nicht allzu schwer zu bewerkstelligen. Falls das Programm aber POKE-Befehle oder Maschinenspracheteile enthält, ist die Anpassung ohne Kenntnis der jeweils anderen Systemadressen kaum möglich.

»PRINT #« abkürzen?

Ich habe das Programm »Datenverwaltung« aus Heft 4/84 abgetippt und es läuft einwandfrei bis auf die Speicherung der Daten. Immer wenn ich »S« (= speichern) drücke und den Kassettenspeicher auf »Aufnahme« schalte, erscheint ein Syntax-Error in Zeile 615.

Hans Mair

In Zeile 615 wird mit dem PRINT #-Befehl in eine Banddatei geschrieben. Sehr wahrscheinlich haben Sie bei der Eingabe dieses Befehls die Abkürzung »?« für PRINT verwendet, was bei PRINT # nicht zulässig ist.

VC 20 erweitern?

Welche Speichererweiterung ist ideal für den VC 20?

Harald Janning

Der VC 20 hat ja leider die unangenehme Eigenschaft, daß Video- und Farbspeicher je nach Speicherausbaueinheit verschiedene Adressen im RAM einnehmen. Ideal erscheint daher die Verwendung einer Erweiterungsplatine mit mehreren Steckplätzen, so daß man zwischen den verschiedenen Speichererweiterungen umschalten kann. Eine andere Lösung ist die Verwendung einer 64 KByte RAM-Karte. Zwar kann der VC 20 mit soviel Speicher eigentlich gar nichts anfangen, aber diese Karte kann alle denkbaren Speichererweiterungen simulieren. Man spart sich so das lästige Umstecken der einzelnen Erweiterungen.

Eine solche 64 KByte RAM-Karte bietet zum Beispiel die Firma Roos Elektronik, Kleiner Markt 7, in 4190 Kleve an.

Maschinensprache laden?

Wie kann ich mit dem Commodore 64 Maschinensprache laden? Welche Hard- oder Software fehlt mir dafür?

Gundolf Plischke

Maschinenspracheprogramme werden entweder mit »LOAD "Name",1,1« (von Kassette) oder mit »LOAD "Name",8,1« (von Diskette) geladen.

Die Sekundäradresse »1« veranlaßt den Computer, das Programm nicht wie üblich an den Basic-Anfang zu verschieben, sondern es an seine originale Startadresse zu laden.

Autostart

Wie bringe ich meine Programme dazu, daß sie nach dem Laden von Diskette oder Kassette automatisch starten?

Carsten Bruch

Beim Laden von Datensette ist ein Autostart sehr einfach: Drücken Sie gleichzeitig »Shift« und die »RUN/STOP«-Taste. Es wird das nächste Programm von Kassette gelesen und sofort gestartet. Beim Diskettenbetrieb empfiehlt es sich, zunächst das DOS 5.1 von der Demo-Diskette zu laden. Der DOS-Befehl »1« (Hochpfeil), gefolgt von einem Programmnamen, lädt das entsprechende Programm von Diskette und startet es automatisch.

So wertet man Formeln in Basic-Programmen aus

Frage: „Wie wertet man Basic-Programme aus?“

Ausgabe: 64/5

Name: Steffen Roehn

Die Routine ab Adresse 44446/\$AD9E (nicht \$AD9F!) wertet zwar sowohl numerische als auch Stringausdrücke aus, jedoch muß beim Auswerten eines Strings nach dessen Bearbeitung derselbe auch wieder aus dem Stringstack entfernt werden, da dieser sonst überläuft (daher auch der »FORMULA TOO COMPLEX«). Dies erfolgt durch Aufruf der Routine »FRESTR« ab Adresse 46755/\$B6A3, wobei gleichzeitig auch geprüft wird, ob es sich bei dem gelesenen Ausdruck auch um einen String handelt. Erklärungen zur Formelauswertung und anderem, was dazu gehört, befinden sich außerdem im »Interface Age Systemhandbuch zum Commodore 64« auf den Seiten 10 bis 22.

Ralph Babel

Entfernungsberechnung mit VC 20

Frage: „Wie kann ich mit dem VC 20 die Entfernung zwischen zwei Koordinaten auf der Oberfläche der Erde berechnen?“

Ausgabe: 64/5

Name: Harald Lang

Ich habe ein solches Programm für C 64 geschrieben. Es läßt sich leicht an den VC 20 anpassen. Interessenten können sich an mich wenden.

Info: Alfred Born, Hermannstr. 10, 1000 Berlin 64, Tel. 030/621 9828

Bildschirm horizontal scrolle

Frage: „Kann ich mit dem C 64 den Bildschirm horizontal scrolle lassen?“

Ausgabe: 64/5

Name: Andres Linz

Das Scrolling (insbesondere Softscrolling) ist im »Interface Age Systemhandbuch zum Commodore 64« auf den Seiten 60 bis 64 ausführlich erklärt. R. Babel

Drucker/Floppy-Einschalt-Test

Frage: „Mit welchen Befehlen kann ich überprüfen, ob der Drucker oder die Floppy eingeschaltet ist?“

Ausgabe: 64/5

Name: Ernst Jeschke

Eine Überprüfung, ob ein Drucker am seriellen Bus angeschlossen ist, läßt sich beim Commodore 64 durch folgende Befehle erreichen:

```
100 POKE 768, 185
110 OPEN 1, 4: PRINT #1
120 POKE 768, 139
```

Da so jegliche Fehlermeldung unterdrückt wird, kann nun durch Abfrage der Statusvariablen »ST« geprüft werden, ob die Leerzeile (oder jeder beliebige andere Text wie zum Beispiel nicht druckende Steuerzeichen!) vom Drucker angenommen wurde. Somit kann verhindert werden, daß ein Programm auf durch »DEVICE NOT PRESENT« abgebrochen wird. Ein Test auf Vorhandensein einer Diskettenstation kann durch entsprechende Programmänderung erfolgen.

Ralph Babel

Um abzufragen, ob der Drucker angeschaltet ist, muß man eine Probezeile an den Drucker schicken. Dies funktioniert nur, falls noch ein anderes Gerät am seriellen Port angeschlossen ist (Floppy, Plotter):

```
10 OPEN 4,4
20 PRINT #4, "—"
30 PRINT #4
40 CLOSE 4
```

2. Möglichkeit: Die Speicherstelle 186 enthält die Adresse des

Gerätes, mit dem zuletzt Daten ausgetauscht wurden. Also 8 bei Disk, 1 bei Datensette und 4 nach dem Drucken. Dies ist in Ladeprogrammen gut verwendbar.

```
10 A = PEEK(186)
20 IF B = 0 THEN B = 1: LOAD "NAME", A, 1
30 SYS Startadresse
```

Somit ist kein Umschreiben von Ladeprogrammen von Floppy auf Recorder oder umgekehrt mehr nötig. Guido Stapel

Schachprogramme

Frage: „Kennt jemand ein spielstarkes Schachprogramm für den C 64?“

Ausgabe: 64/5

Name: Peter Jugl

Außer den genannten Schachprogrammen Grandmaster und Sargon II gibt es, soweit mir bekannt, nur noch zwei spielstärkere Programme:

1. CHESS 7.0 von ODESTA (auf Diskette)

Features: Spiele können auf Diskette abgespeichert und wieder eingelesen werden, viele Meisterpartien sind bereits gespeichert, Blindschachvariationen möglich, Zurücknahme, Prüfung eines Feldes auf Bedrohung und so weiter. Preis: zirka 300 Mark.

2. Colossus 2.0 (auf Kassette)

Ähnliche Möglichkeiten wie oben, aber durch die Kassettenspeicherung etwas begrenzt. Preis: zirka 120 Mark. Beide Programme sind im Fachhandel erhältlich.

Guido Stapel
Vermutlich wird Herrn Jugl kein Computerprogramm schlagen, so daß er auf einen Schachcomputer zurückgreifen sollte. Die Spitzenschachcomputer schlagen Computerschachprogramme übrigens recht einfach und überzeugend. Die guten NOVAG-Schachcomputer haben mich persönlich überzeugt, da sie recht spielstark sind, keine Spielfehler machen (jedenfalls keine offensichtlichen) und ständig auf den neuesten Stand gebracht werden können, da sie mit EPROM-Technik arbeiten. Das heißt: Gibt es ein neues, spielstärkeres Programm, so kann man seinen alten Schachcomputer einfach nachrüsten (= austauschen). Detlef Wacker

Vom Bildschirm auf Kassette

Frage: „Ich suche ein Programm, um Daten vom Bildschirm (C 64) auf Kassette speichern und von dort wieder abrufen zu können“

Ausgabe: 64/5

Name: Thomas Mandl

Das Problem, welches sich Herrn Mandl stellt, muß in zwei

Kategorien aufgespalten werden:

1) Abspeichern des Bildschirm-inhaltes 40 x 25 Zeichen:

Wenn nur rechner-spezifische Zeichen (also keine hochauflösende Grafik) abgespeichert werden sollen, so ist dies recht einfach. Eine denkbare Lösung wäre, die 1000 Zeichen (40 x 25) mit der PEEK(X)-Funktion in einen Vektor einzulesen und diesen Vektor dann in ein sequentielles File auf Kassette zu schreiben. Die 1000 Zeichen stehen im Computer ab Adresse 1024 bis Adresse 2023. Mit folgendem Programm können die Zeichen ausgelesen werden:

```
10 DIM A(1000)
20 FOR I = 1 TO 1000
30 A(I) = PEEK(I + 1023)
40 NEXT I
```

Dieser Vektor (A) kann nun einfach auf Kassette geschrieben und jederzeit wieder geladen werden. Mit POKE(X) kann dann der Text/die Grafikzeichen einfach wieder auf den Bildschirm zurückbefördert werden. Eine Erweiterung des obigen Programmes verdeutlicht dies:

```
50 PRINT "SHIFT/CLR HOME"
(Bildschirm löschen)
60 FOR I = 1 TO 1000
70 POKE 1023 + I, A(I)
80 NEXT I
90 END
```

Wenn man noch die Farben der 1000 Zeichen abspeichern will, so muß man zusätzlich noch den Speicherbereich von 55296 bis 56295 auslesen. Das Verfahren ist identisch! So kann immerhin schon eine Blockgrafik (mit Grafikzeichen) in Farbe oder ein Text abgespeichert werden.

2) Abspeichern von hochauflösender Grafik:

Bei hochauflösender Grafik ist das schon etwas umständlicher, denn hier muß man das Bitmuster der Grafik abspeichern. Das Verfahren ist identisch mit jenem, das eine hochauflösende Grafik auf einen Drucker bringt (Hardcopy). Vielleicht schreibt ein anderer Leser für Herrn Mandl ein entsprechendes Programm, ich habe leider keine Zeit dazu, da ich im Abistreß stehe.

Detlef Wacker

Um Daten direkt vom Bildschirm auf Kassette übertragen zu können, muß man in Basic mit Files arbeiten:

```
60000 OPEN 1,1,1,"Bildschirm"
60050 FOR I = 1024 TO 2023
60100 P = PEEK(I):PRINT #1, I
60150 NEXT I: CLOSE 1
```

Nun umgekehrt:

```
60000 OPEN 1,1,0,"Bildschirm"
60050 FOR I = 1024 TO 2023
60100 INPUT #1, P: POKE I, P
60150 NEXT I: CLOSE 1
```

Guido Stapel

Das DOS auf der Demodiskette — noch 2 Befehle

Bei Ihrem Bericht über das DOS 5.1 in der Mai-Ausgabe des 64'ers wurden zwei Befehle des DOS 5.1 nicht behandelt.

Das ist einmal der Befehl: % Name.

Dieser Befehl lädt ein Maschinenprogramm ein, das sonst mit LOAD "Name", 8, 1 geladen werden muß. Nach diesem Befehl braucht auch kein NEW mehr eingegeben werden, wie es sonst nach dem Laden eines Maschinenprogramms nötig ist, um keinen OUT OF MEMORY ERROR zu erhalten.

Der zweite der beiden weiteren Befehle ist >Q.

Er bewirkt, daß das DOS 5.1 verlassen wird. Dies ist dann wichtig, wenn ein anderes Programm in den Bereich geladen werden soll, in dem das DOS 5.1 liegt (hex CC00 bis CF58 = dez 52224 bis 53080). Da die Routine zur Überprüfung der Eingabe in diesem Bereich liegt, würde ein Überschreiben dieser Routine in den meisten Fällen zum Absturz führen.

Das DOS 5.1 kann nach >Q mit SYS 52224 wieder gestartet werden.

Frank Tecker

Wollen Sie antworten?

Wir veröffentlichen auf dieser Seite auch Fragen, die sich nicht ohne weiteres anhand eines guten Archivs oder aufgrund der Sachkunde eines Herstellers beziehungsweise Programmierers beantworten lassen. Das ist vor allem der Fall, wenn es um bestimmte Erfahrungen geht oder um die Suche nach speziellen Programmen beziehungsweise Produkten. Wenn Sie eine Antwort auf eine hier veröffentlichte Frage wissen — oder eine andere bessere Antwort als die hier gelesene — dann schreiben Sie uns doch. Antworten publizieren wir in einer der nächsten Ausgaben. Bei Bedarf stellen wir auch den Kontakt zwischen Lesern her.

Der Nach

Die Überschrift wirft sofort eine Frage auf: Nachfolger von wem? Vom VC 20 oder vom C 64?

In den letzten Heften wurde bereits ausführlich über den Commodore 264 berichtet und dabei ist klar geworden, daß der 264 den C 64 nicht ablösen soll. Der 264 ist etwas weiter weg vom Hobby-Computer: etwas weniger Grafik- und Sound-Möglichkeiten, dafür besseres Basic und komfortablere Handhabung, ein Computer weniger zum Spielen als mehr zum ernsthaften Arbeiten. Um den neuen C 116/C 16 in die Gesamtpalette der kleineren Commodore-Systeme einordnen zu können, stellt man am besten einmal die Eigenschaften und Fähigkeiten aller vier Computer gegenüber. Fangen wir mit den alten Bekannten an.

VC 20

Als er auf den Markt kam, wurde er innerhalb kürzester Zeit zum Markttrenner. Sein ausgezeichnetes Preis/Leistungsverhältnis war konkurrenzlos. Die Commodore-Strategie (computers not for the classes but for the masses) kam voll zum Tragen. In der Grundversion 3 KByte (heute lacht man fast darüber), eine fast professionelle Tastatur, ausbaufähig, eine schnell wachsende Menge an Software: der VC 20 wurde zum Liebling der deutschen Computerszene. Die maximal mögli-



Escape-Taste

viele Herzen höher schlagen. Wer bisher noch gewartet hatte mit der Anschaffung eines Computers, war jetzt überzeugt, die richtige Wahl treffen zu können. Und seine Fähig-

Stromanschluß

Reset-Taste

Serielle Schnittstelle

Datasette

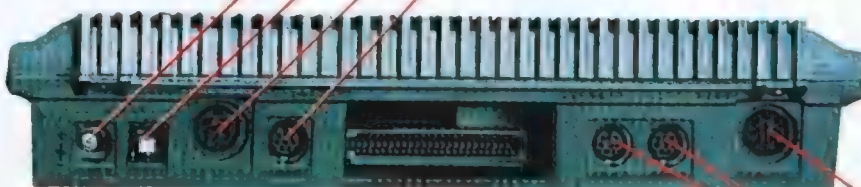


Bild 3. Die Anschlüsse des C 116. Sie dürften denen des C 16 entsprechen.

che Anzahl von 22 Zeichen pro Zeile und 23 Zeilen am Bildschirm störten damals noch niemanden. Die Auflösung von 160 x 160 Punkten erlaubte die Darstellung befriedigender Grafiken. Der Preis von damals 800 Mark, heute zirka 300 Mark, ließ den VC 20 für viele erschwinglich werden.

Commodore 64

Es dauerte aber nicht lange, da kam der C 64 auf den Markt. Er ließ

keiten ließen viele andere vor Neid erblassen: 64 KByte RAM, davon zirka 38 KByte frei für Basic-Programme, ungefähr 52 KByte für Maschinensprache-Programme, eine Grafikauflösung von satten 320 x 200 Punkten, 16 Farben und schon fast professionelle Synthesizer-Eigenschaften. Und der Preis stimmte auch. Was wollte man mehr? Das spartanische Basic und die etwas mühselige Art, Grafik und Sound zu erzeugen, schreckte fast niemanden; im Gegenteil: eben weil keine starren Befehle die Mög-

Ein etwas ungutes Gefühl beschlich uns, als wir hörten, daß einer der neuen Commodore-Computer eine Gummitastatur besitzen sollte.

Commodore mag ein ähnliches verspürt haben, denn zur Hannover-Messe wurde im letzten Moment das Ruder herumgeworfen: Der Gummicommodore C 116

Videoeingang

Joystick-Anschlüsse

wandert zurück in die Schublade und an seine Stelle tritt der C 16, mit dem gleichen Innenleben, aber mit der bekannt guten Tastatur des C 64.

folger?

Cursor-Tasten

HELP-Funktionstaste

schreibt, die mehr als 38 KByte benötigen, sollte einmal seinen Programmierstil überprüfen...). Auch die Benutzung fest eingebauter Programme (Textverarbeitung, Kalkulationsprogramme oder andere Programmiersprachen) auf Knopf-

◀ Bild 2. Der Commodore C 16 ist ins Gehäuse des C 64/VC 20 gepackt. Nur die Anordnung der Tasten wurde geändert.

Ausgaben auf dem Bildschirm. Dies alles, in Verbindung mit dem komfortablen Basic 3.5, macht den Commodore 264 interessant für den anspruchsvollen Heim-Anwender.

C 16

Dieser Computer ist sozusagen eine verkleinerte Ausgabe des C 264; oder man kann auch sagen, sein kleiner Bruder (Bild 2). Er besitzt die gleiche komfortable Basic-Version 3.5 und auch die gleichen Grafik- und Sound-Möglichkeiten. Eine eingebaute (built-in) Software wird es jedoch nicht geben und ein Windowing wird auch nicht möglich sein. Das liegt an der begrenzten Speicherkapazität. Diese beträgt nur 16 KByte RAM. Der C 16 kennt, ebenso wenig wie der 264 und der VC 20, keine Sprites, diese für Spiele so interessanten, frei beweglichen und

Home/Clr

Return-Taste

Cursortaste ähnlich einem Joystick

definierbaren Grafikgebilde des C 64. Allerdings beträgt seine Grafikauflösung 320 x 200 Punkte, einzeln ansteuerbar und in 16 Farben mit je acht Helligkeitsstufen über Basic-Befehle leicht ansprechbar. Ebenso wie der C 64 und der C 264 kann der C 16 40 Zeichen pro Zeile darstellen, das übliche bei heutigen Homecomputern. Die Sound-Fähigkeiten beschränken sich auf zwei Tongeneratoren, einer für einfache Töne, der andere für Geräusche wie Rauschen, Knallen, Donnern und so weiter, selbst gegenüber den Möglichkeiten des VC 20 enttäuschend mager (siehe auch Tabelle 1). Der

Escape-Taste

lichkeiten des C 64 und die Fähigkeiten des Programmierers einschränken, kann jeder etwas dazu lernen. Selbst der Profi hat es schwer, an die Grenzen des Machbaren zu kommen. Auch beim C 64 wächst die Anzahl der verfügbaren Software ins Unübersehbare. Selbst CP/M-Software (zum Beispiel Wordstar) gibt es schon für den C 64.

C 264

Dieser neue Commodore ist ohne Zweifel der eleganteste. Sein Gehäuse ist sehr kompakt, nicht so klobig wie die anderen.

Seine Tastatur ist flach und erlaubt auch längeres ermüdungsfreies Arbeiten. Ein freier Speicherplatz von zirka 60 KByte ermöglicht das Entwerfen großer Programme (allerdings: Wer Basic-Programme

▲ Bild 1. Der C 116 kommt nicht auf den deutschen Markt.

druck ist neu. Selbst ein Windowing ist möglich, die Darstellung mehrerer voneinander unabhängiger



Reset-Taste

Der Nachfolger?

C 16 wurde in das Gehäuse des VC 20/C 64 gesteckt. Die Tastatur ist von der Mechanik her die gleiche, lediglich die Tastenbelegung ist geändert worden. Die Cursortasten befinden sich jetzt oben rechts und es gibt nun vier davon, für jede Richtung eine. Die RESTORE-Taste wird jetzt mit CLR/HOME belegt. Die Taste »Pfeil nach links« entfällt und wird ersetzt durch die ESC-Funktion.

Abgesehen von der Tastatur ist der C 116 mit seinen Gummitasten übrigens identisch mit dem C 16 (Bild 1+3+4). Der C 116 wird auf dem deutschen Markt nicht erhältlich sein.

Fazit

Wer sich einen neuen Computer kaufen will, ist mit dem C 64 bestimmt um einiges besser beraten. Der C 64 hat so eine gewaltige Verbreitung erreicht, daß sein Vorsprung auf dem Software-, Zubehör- und auf dem Literaturmarkt selbst innerhalb eines Jahres nicht zu schlagen sein wird, wenn überhaupt. Reicht das Taschengeld nicht ganz, ist aber der C 16 eine annehmbare Alternative. Er soll um die 300 Mark kosten und leistet einiges mehr als der VC 20.

Wer jedoch einen C 64 besitzt und auf ein größeres System umsteigen will, sollte lieber noch etwas warten. Gegen Ende des Jahres dürfte es auch von Commodore einen IBM-kompatiblen Computer geben, mit entsprechender Software, mit mehr als 64 KByte Speicherplatz und einem schnellen 16-Bit-Prozessor.

```
100 REM BENCHMARK 1
110 A=TI
200 PRINT"S"
400 FOR K=1 TO 1000
444 NEXTK
448 B=TI
450 PRINT(B-A)/60
500 PRINT"E"
600 PRINT :LIST
READY.
```

Benchmark 1

```
100 REM BENCHMARK 2
110 A=TI
200 PRINT"S"
300 K=0
400 K=K+1
444 IFK<1000 THEN 400
448 B=TI
450 PRINT(B-A)/60
500 PRINT"E"
600 PRINT :LIST
READY.
```

Benchmark 2

```
100 REM BENCHMARK 5
110 A=TI
120 PRINT"S"
130 K=0
140 K=K+1
150 C=K/2*3+4-5
155 GOSUB 195
160 IFK<1000 THEN 140
170 B=TI
180 PRINT(B-A)/60
190 PRINT"E"
195 RETURN
200 PRINT :LIST
READY.
```

Benchmark 5

```
100 REM BENCHMARK 6
110 A=TI
120 PRINT"S"
130 K=0
135 DIM M(5)
140 K=K+1
150 C=K/2*3+4-5
155 GOSUB 195
156 FOR L=1 TO 5
157 NEXT L
160 IFK<1000 THEN 140
170 B=TI
180 PRINT(B-A)/60
190 PRINT"E"
195 RETURN
200 PRINT :LIST
READY.
```

Benchmark 6

Commodore C 16

Speicher: 32-KB-ROM-Betriebssystem und Basic 3.5-Interpreter 16 KB RAM

Prozessoren: — 7501 Mikroprozessor mit 0.89-1.76 MHz Taktfrequenz

Grafik: — 40 Spalten x 25 Zeilen — 121 Farbtöne (16 Farben mit 8 Helligkeitsstufen) — Groß- und Kleinschreibung, Grafiksymbole

— Hochauflösende Grafik 320 x 200 Punkte mit voller Basic-Unterstützung — geteilter Bildschirm (Text/Grafik gemischt)

Synthesizer: 1 Tongenerator und 1 Rauschgenerator — 8 Lautstärkepegel

Tastatur: 67 Tasten, 8 programmierbare Funktionstasten, Cursor-Steuerpult

Ein-/Ausgabe: Serielle Schnittstelle, Kassettenport, Erweiterungsport, 2 Joystickanschlüsse, Video- und TV-Ausgang

Software: Basic 3.5-Interpreter (über 75 Befehle)

Maße: 40 cm x 7 cm x 21 cm (BxHxT)

Tabelle 1. Einige technische Daten des C 16 auf einen Blick

Geschwindigkeitsvergleiche

Ich habe mit den Computern VC 20, C 116, C 64 und C 264 einen Geschwindigkeitsvergleichstest durchgeführt. In den nebenstehenden Listings sind die einzelnen Benchmarks abgedruckt. Diese Benchmarktests (siehe auch Tabelle 2) beschränken sich auf die Basic-Befehle, die alle Computer gemeinsam haben. Mit einigen Befehlen des neuen Commodore Basic 3.5 könnten sich die Ausführungszeiten einiger Tests vielleicht verbessern. Dieser direkte Vergleich hat deshalb nur einen eingeschränkten Aussagewert.


```

100 REM BENCHMARK 3
110 A=TI
200 PRINT"S"
300 K=0
400 K=K+1
410 C=K/K*K+K-K
444 IFK<1000 THEN 400
448 B=TI
450 PRINT(B-A)/60
500 PRINT"E"
600 PRINT :LIST
READY.

```

Benchmark 3

```

100 REM BENCHMARK 4
110 A=TI
200 PRINT"S"
300 K=0
400 K=K+1
420 C=K/2*3+4-5
444 IFK<1000 THEN 400
448 B=TI
450 PRINT(B-A)/60
500 PRINT"E"
600 PRINT :LIST
READY.

```

Benchmark 4

```

100 REM BENCHMARK 7
110 A=TI
120 PRINT"S"
130 K=0
140 DIM M(5)
150 K=K+1
160 C=K/2*3+4-5
170 GOSUB 240
180 FOR L=1 TO 5
185 M(L)=A
190 NEXT L
200 IFK<1000 THEN 150
210 B=TI
220 PRINT(B-A)/60
230 PRINT"E"
240 RETURN
250 PRINT :LIST
READY.

```

Benchmark 7

```

100 REM BENCHMARK 8
110 A=TI
200 PRINT"S"
300 K=0
400 K=K+1
420 C=K+2
422 D=LOG(K)
426 E=SIN(K)
444 IFK<1000 THEN 400
448 B=TI
450 PRINT(B-A)/60
500 PRINT"E"
600 PRINT :LIST
READY.

```

Benchmark 8

den das erste Mal vom US-Magazin »Kilobaud« veröffentlicht im Jahre 1977. Lediglich Benchmark 8 wurde später hinzugefügt.

Abgesehen von Benchmark 1, das eine einfache FOR-NEXT-Schleife ausführt und Benchmark 8, das die Funktionen Potenzieren, Logarithmieren und die Sinus-Funktion ausführt, sind die restlichen Benchmarks so strukturiert, daß durch Subtraktion der Zeit des vorhergehenden Tests sich die neu hinzugefügte Funktion isolieren läßt. So kann man zum Beispiel durch Abziehen Benchmark 4 von Benchmark 5 die Zeit für 1000 Gosub/Return feststellen.

Zu diesen Benchmarktests noch ein paar Bemerkungen.

Benchmarktests können nie eine allgemeine Aussage treffen. Schwächen auf einem Gebiet werden durch Stärken in anderen Bereichen aufgehoben. Um ein abgerundetes Bild zu erhalten, muß auch die Bildschirmausgabe oder der Umgang mit Strings berücksichtigt werden. Die Geschwindigkeit bei der Grafikerstellung ist wieder für andere ein wichtiges Kriterium. Viele Handicaps lassen sich aber auch durch Maschinensprache-Routinen umgehen. (gk)

Lichtgriffel mit Software

Den hochempfindlichen Lichtgriffel von Madison Computer gibt es jetzt auch mit der entsprechenden Software wie QB Graphics, Tic Tac Toe und einigen Utilities für den VC 20 oder Commodore 64 zu einem Preis von 50 Dollar. Diese Programme eignen sich hervorragend für Kinder im Vorschulalter, da, sobald ein Programm geladen ist, keine einzige Eingabe mehr über die Tastatur zu erfolgen braucht. Lediglich durch Zeigen mit dem Lichtgriffel auf den Bildschirm werden Zahlen, Buchstaben oder Farben ausgewählt. Auch eigene Programme sollen, nach entsprechender Lektüre des Handbuchs erstellt werden können.

Den Lichtgriffel mit der Software auf Diskette oder Kassette gibt es bei Madison Computer, 1825 Monroe Street, Madison, WI.

Benchmark	VC 20	C 116	C 64	C 264
1	1.15	1.48	1.42	1.98
2	8.13	8.06	9.72	10.77
3	15.43	15.30	18.72	20.52
4	16.88	15.75	20.15	20.98
5	18.36	17.93	21.85	23.95
6	27.33	29.21	32.60	39.18
7	42.87	46.08	51.13	61.75
8	97.83	83.78	116.63	111.63

Tabelle 2. Geschwindigkeitsvergleich. Alle Angaben in Sekunden. Zur Zeitmessung wurde die rechnerinterne Uhr (TI) genommen.

Folgende Gründe könnte es geben für die unterschiedlichen Ausführungszeiten: Der VC 20 ist prinzipiell der Schnellste, weil er mit einer etwas höheren Taktfrequenz läuft. Der C116 ist schneller als der C264,

weil sein Betriebssystem nicht ganz so umfangreich ist wie das des C264. Deshalb müssen nicht so viele Betriebssystem-Routinen abgefragt und ausgeführt werden.

Diese Tests (siehe Listings) wur-

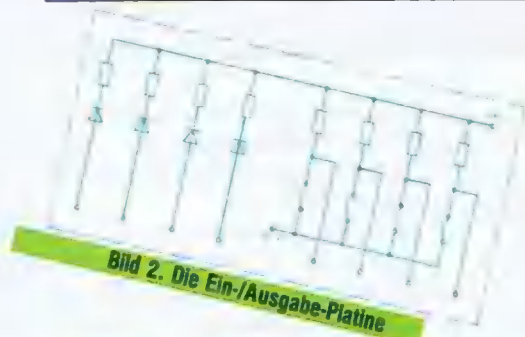


Bild 2. Die Ein-/Ausgabe-Platine

Kopplung zwischen zwei VC 20

Wenn Sie auf ein größeres System umsteigen, dann verschleudern Sie nicht Ihren alten VC 20. Nutzen Sie statt dessen seine experimentellen Möglichkeiten, beispielsweise für die bitparallele Datenübertragung. Der folgende Artikel bringt dazu einige Anregungen und eine über zwei VC 20 zu spielende Knebelerei.

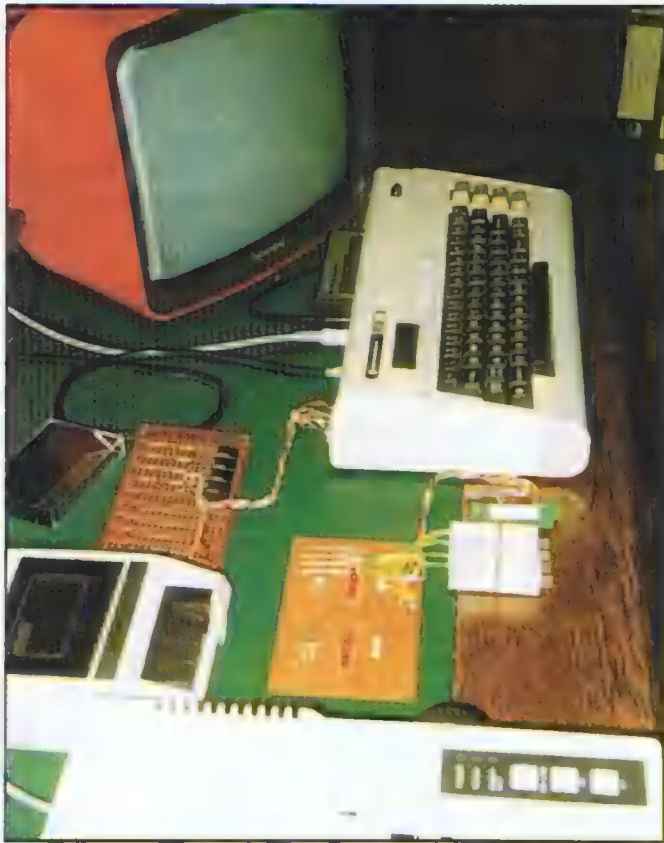


Bild 3. Zwei verschiedene Ein-/Ausgabe-Platinen

Auf der Rückseite des VC 20 befindet sich unter anderem der sogenannte USER-PORT mit seinen 24 Anschlüssen. Die Pin-Belegung mit Blick auf den Port zeigt Bild 1. Für die folgenden Experimente werden neben einem (beliebigen) Masseanschluß (GND) und der herausgeführten Betriebsspannung von +5 V (Pin 2) die acht Ein- beziehungsweise Ausgänge des Ports B, das heißt die Anschlüsse C bis L benötigt. Die übrigen Anschlüsse sind zum Teil nicht beliebig für Experimente frei, da sie für andere Zwecke mit benötigt werden. So führen die Pins 4 bis 6 zum Joystickanschluß, Pin 7 ist für den Lichtgriffel, Pin 8 für den Kassettenschalter und Pin 9 ist ein gepufferter Serienaussgang. Zwei weitere Anschlüsse, Pin B und M mit den Bezeichnungen CB 1 und CB 2 sind sogenannte quittierende Leitungen (Handshake-Leitungen).

Über Port B kann ein 8 Bit breites

Wort ausgegeben beziehungsweise eingelesen werden. Die Kontrolle des Ports erfolgt über den Baustein 6522, der unter anderem die Richtung des Datenflusses mit Hilfe des Datenrichtungsregisters (37138) festlegt. Nach dem Einschalten des Computers sind alle Leitungen von Port B Eingänge.

Beim Ausgeben soll der Computer an die Pins C bis L (mit den Bezeichnungen PB 0 bis PB 7) eine logische 1 (entspricht +5 Volt) beziehungsweise eine Null (0 Volt) legen. Dazu sind zwei Schritte notwendig: a) Der ganze Port beziehungsweise einzelne Leitungen von ihm müssen als Ausgänge programmiert werden. Der entsprechende Basic-Befehl heißt POKE 37138,X.

X = 1 : PB 0 ist Ausgang

X = 2 : PB 1 ist Ausgang

X = 4 : PB 2 ist Ausgang (und so weiter bis X = 128)

X = 3 : PB 0 und PB 1 sind Ausgänge
X = 5 : PB 0 und PB 2 sind Ausgänge (und so weiter bis X = 255).

b) Auf die als Ausgänge vorprogrammierten Leitungen wird 1 beziehungsweise 0 gegeben. Der entsprechende Befehl heißt POKE 37136,Y.

Y = 0 : auf allen Leitungen liegt 0.

Y = 1 : nur auf PB 0 liegt 1

Y = 2 : nur auf PB 1 liegt 1

Y = 4 : nur auf PB 2 liegt 1 (und so weiter bis Y = 128)

Y = 3 : auf PB 0 und PB 1 liegt 1 (und so weiter bis Y = 255).

Auch das Einlesen von Port B erfolgt in zwei Schritten:

a) Es muß darauf geachtet werden, daß die anzusprechenden Leitungen nicht als Ausgänge geschaltet sind. Der entsprechende Befehl heißt ebenfalls POKE 37138,X. Für X ist jedoch die zu 255 komplementäre Binärzahl einzusetzen:

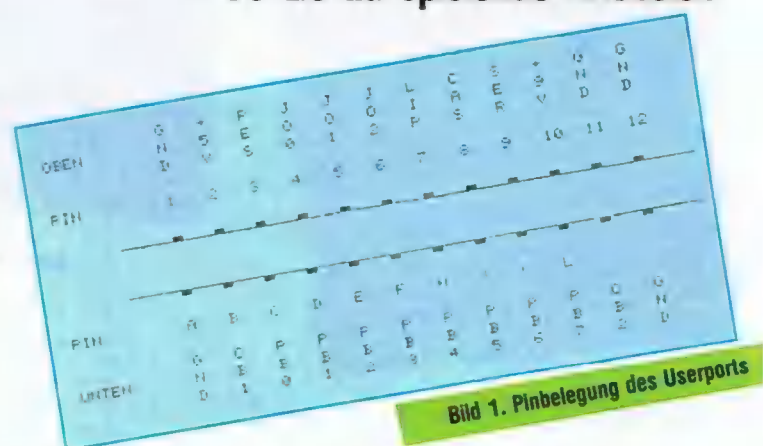


Bild 1. Pinbelegung des Userports

X = 255 : keine Leitung ist Eingang (das heißt alle sind Ausgänge)
 X = 254 : nur PB 0 ist Eingang
 X = 240 : PB 0 ... PB 3 sind Eingänge (Rest Ausgänge)
 X = 15 : PB 4 ... PB 7 sind Eingänge
 X = 0 : alle Leitungen sind Eingänge.

b) Das gesamte angelegte Wort (einschließlich eventuell ausgegebener Einsen) wird eingelesen. Der entsprechende Basic-Befehl heißt PEEK (37136). Dies ergibt Zahlenwerte zwischen 0 und 255. Der Wert 255 bedeutet, daß an allen acht Pins eine logische 1, das heißt +5 V liegt. Sollen einzelne Leitungen abgefragt

Programmen steht P 0 statt PB 0 und so weiter.

Für die folgenden Experimente empfiehlt es sich, eine einfache Ein-/Ausgabeplatine zu bauen, die

beteil des Computers) legen Taster beziehungsweise Schalter die Eingänge auf +5 V. Bei offenen Tastern taucht ein Problem auf, da der Computer ebenfalls eine 1 einliest. Des-

Bild 5. Logische Struktur des Knobelspiels

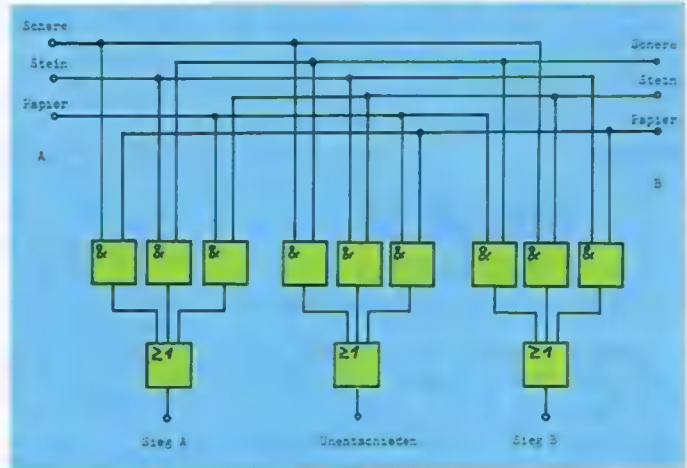
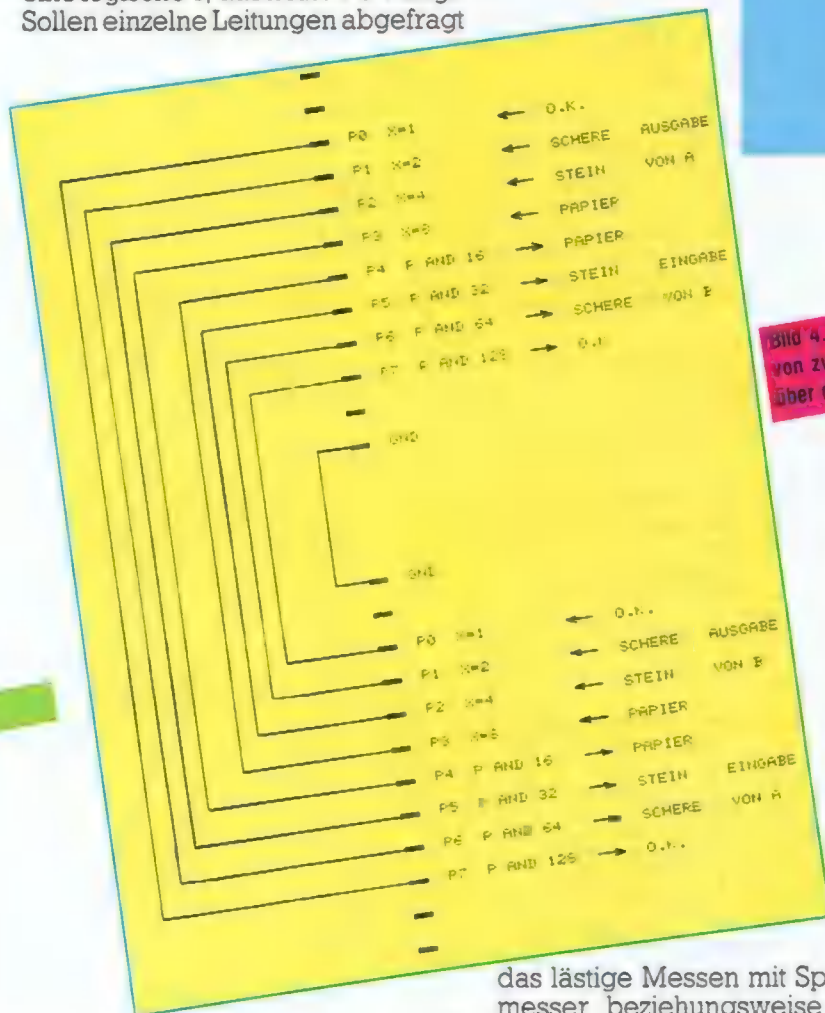


Bild 4. Kopplung von zwei VC 20 über den Userport.



werden, so gewinnt man ihre Werte durch logischen Vergleich, den man am besten in einem Unterprogramm durchführt:

```
1000 P = PEEK (37136)
1010 P 0 = - ((P AND 1) = 1)
1020 P 1 = - ((P AND 2) = 2)
1030 P 2 = - ((P AND 4) = 3)
```

```
1080 P 7 = - ((P AND 128) = 128)
1090 PRINT P 0; P 1; ... P 7
```

Das Unterprogramm führt bitweise einen logischen UND-Vergleich durch und prüft, ob das jeweilige Bit mit 1 oder 0 belegt ist.

Anmerkung: In den folgenden

das lästige Messen mit Spannungsmesser beziehungsweise Logiktester ersetzt. Einen entsprechenden Schaltungsvorschlag zeigt Bild 2.

Die Schaltung ist so ausgelegt, daß die vier Leitungen PB 0 bis PB 3 als Ausgänge und die vier Leitungen PB 4 bis PB 7 als Eingänge für den Computer verwendet werden. Selbstverständlich sind andere Aufteilungen möglich.

Der Anzeigeteil der Platine (entspricht dem Ausgabeteil des Computers) ist mit vier roten LEDs mit Vorwiderstand bestückt. Bei 200 Ohm bleibt selbst bei acht LEDs der Gesamtstrom unter den zulässigen 100 mA.

Im Ausgabeteil der Platine (Einga-

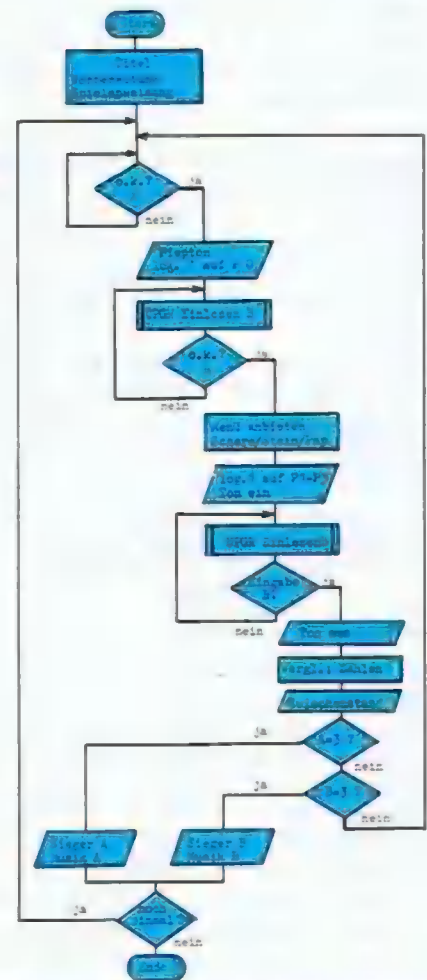


Bild 6. Flußdiagramm für das Knobelspiel

halb wird durch die 1 kOhm-Widerstände das Potential auf logisch 0, das heißt auf zirka 0 V herabgezogen.

Für den Aufbau eignet sich fast jede Experimentierplatine im Europaformat mit Ätzinseln. Die Taster sollten prellfrei sein. Für den User-Port empfiehlt sich der Doppelstecker

CINCH 251-12-90-160 Typ 50 24 EE-30 (erhältlich zum Beispiel bei Hofacker). Für die Zwischenverbindungen genügen bei kurzen Strecken ein mehradriges Band beziehungsweise 10 Einzeldrähte. Bei größeren Entfernungen (ab zirka 2 m) empfiehlt sich ein abgeschirmtes Kabel (Mantel auf GND). Bild 3 zeigt zwei verschiedene Baumuster der Platine.

Software zur Ein-/Ausgabe

Einen Programmvorschlag für das Ausgeben beziehungsweise Einlesen zeigt Listing 1. Da im obigen Beispiel P0 bis P3 als Ausgänge und P4 bis P7 als Eingänge geschaltet werden sollen, wird zunächst der Befehle POKE 37138, 15 gegeben werden ($15 = 8 + 4 + 2 + 1$), das heißt Bit 1 bis 4 sind Ausgänge, der Rest Eingänge.

Das Programm bietet in einem Menü zwei Teilprogramme an: Beim Ausgeben wird durch Betätigen der Tasten Z, X, C, V jeweils eine einzelne 1 auf den Computerport gegeben, was durch das Leuchten der entsprechenden LED angezeigt wird.

Beim Einlesen können einzelne oder alle Taster betätigt werden. Der Computer zeigt sowohl PEEK (37136) als Summe als auch die einzelnen Bitwerte an.

Kopplung von zwei VC 20

Will man statt der beschriebenen Ein-/Ausgabeplatinen einen zweiten Computer verwenden, so muß er mit einer neunadrigen Leitung (für 8 Bit und GND) angeschlossen werden. Mit Rücksicht auf das später zu beschreibende Spiel sind im folgenden vier Leitungen für die eine Datenrichtung, die vier anderen für die umgekehrte Richtung vorgesehen. Außerdem sind die Leitungen untereinander gekreuzt (das heißt Bit 1 von Computer A ist verbunden mit Bit 8 vom Computer B). Dies erlaubt es später, für das Computerspiel in beiden Computern das exakt gleiche Programm verwenden zu können.

Die komplette Verbindung zeigt Bild 4. Es werden die gleichen Stecker und Leitungen wie oben benutzt. Für Testzwecke kann das Programm von Listing 2 benützt wer-

```

10 REM KNOBELSPIEL
20 PRINT "Q"
30 PRINT "  KNOBELSPIEL  "
40 PRINT:PRINT " SCHERE STEIN PAPIER " :PRINT:PRINT
50 PRINT:PRINT "      AUTOR      ALBRECHT LANGENSTEIN"
60 FOR T=1 TO 9999:NEXT
70 POKE 37138,15
80 POKE 37136,0
90 A=0:B=0
100 REM VORBEREITUNG
110 PRINT "ACHTUNG!"
115 PRINT "-----"
120 PRINT "PRUEFEN SIE, OB DAS VERBINDUNGSKABEL STECKT." :PRINT
130 PRINT "SONST ERST BEIDE " :PRINT
135 PRINT "-- RECHNER AUSSCHALTEN"
140 PRINT "-- KABEL EINSTECKEN "
145 PRINT "-- PROGRAMM NEU LADEN." :PRINT
150 PRINT "MONITOR GEEIGNET AUFSTELLEN" :PRINT
155 PRINT "BEI BILDSTREIFEN EVT. UHF-MODULATOREN DREHEN"
160 PRINT "LAUTSTÄRKE EINSTELLEN"
165 POKE 36878,15:POKE 36875,219
170 PRINT " WENN ALLES OK,  LEERTASTE DRUECKEN!"
180 GET$:IF A$="" THEN 180
190 IF A$=" " THEN 200
195 GOTO 170
200 REM ANWEISUNG
205 PRINT "Q":PRINT "SPIELANWEISUNG:"
210 PRINT "-----"
220 PRINT "WAEHLN SIE SCHERE, STEIN ODER PAPIER!" :PRINT
230 PRINT "SCHERE GEWINNT GEGEN PAPIER." :PRINT
240 PRINT "PAPIER GEWINNT GEGEN STEIN." :PRINT
250 PRINT "STEIN GEWINNT GEGEN SCHERE." :PRINT
260 PRINT "SONST UNENTSCHEIDEN." :PRINT
270 PRINT "GESAMTSIEGER IST, WER ZUERST 3 EINZELSPIELE GEWONNEN HAT!"
300 REM START
305 POKE 36878,0
310 P0=0:P1=0:P2=0:P3=0
315 PRINT
320 PRINT "WENN OK, TASTE F1 "
330 GET$:IF A$="" THEN 330
340 IF A$=CHR$(133) THEN 350
345 GOTO 320
350 POKE 36878,15:POKE 36875,237
352 FOR T=1 TO 150:NEXT
355 POKE 36878,0
360 P0=1:X=1:POKE 37136,X
365 PRINT "Q":PRINT "AUF GEGNER WARTEN!"
370 GOSUB 1000
380 IF P7=1 THEN 400
390 GOTO 370
400 REM WAHL A
410 PRINT "Q"
415 PRINT:PRINT " BITTE WAEHLN!" :PRINT:PRINT
420 PRINT " F3...SCHERE " :PRINT:PRINT
422 PRINT " F5...STEIN " :PRINT:PRINT
424 PRINT " F7...PAPIER " :PRINT:PRINT
426 PRINT:PRINT "ENTSPR. TASTE DRUECKEN!"
430 GET$:IF A$="" THEN 430
435 IF A$=CHR$(134) THEN 460
440 IF A$=CHR$(135) THEN 470
445 IF A$=CHR$(136) THEN 480
450 GOTO 400
460 P1=1:X=2:P$="SCHERE"
465 POKE 37136,X:GOTO 500
470 P2=1:X=4:P$="STEIN "
475 POKE 37136,X:GOTO 500
480 P3=1:X=8:P$="PAPIER"
485 POKE 37136,X:GOTO 500
500 REM EIN B
505 POKE 36878,15:POKE 36875,219
510 GOSUB 1000
520 IF P4=1 OR P5=1 OR P6=1 THEN 540
530 GOTO 510
540 POKE 36878,0
500 REM VERGLEICH
601 IF P6=1 THEN 605
602 IF P5=1 THEN 606
603 IF P4=1 THEN 607
605 Q$="SCHERE":GOTO 610
606 Q$="STEIN ":GOTO 610
607 Q$="PAPIER":GOTO 610
610 IF P1=1 AND P6=1 THEN 710:REM REMIS
620 IF P2=1 AND P5=1 THEN 710
630 IF P3=1 AND P4=1 THEN 710
640 IF P1=1 AND P4=1 THEN 740:REM SIEG A
650 IF P2=1 AND P6=1 THEN 740
660 IF P3=1 AND P5=1 THEN 740

```



```

670 IF P1=1 AND P5=1 THEN 770:REM SIEG B
680 IF P2=1 AND P4=1 THEN 770
690 IF P3=1 AND P6=1 THEN 770
700 REM ERGEBNIS
710 REM REMIS
712 PRINT"Q":PRINT"SIE HATTEN ";P$
714 PRINT:PRINT"IHR GEGNER ";Q$
720 PRINT:PRINT" ALSO UNENTSCHIEDEN"
730 GOTO 800
740 REM SIEGER SELBST
742 PRINT"Q":PRINT"SIE HATTEN ";P$
744 PRINT:PRINT"IHR GEGNER ";Q$
750 PRINT:PRINT"SIE HABEN GEWONNEN! "
760 A=A+1:GOTO 800
770 REM SIEGER GEGNER
772 PRINT"Q":PRINT"SIE HATTEN ";P$
774 PRINT:PRINT"IHR GEGNER ";Q$
780 PRINT:PRINT"GEGNER HAT GEWONNEN!"
790 B=B+1:GOTO 800
800 REM ZAEHLEN
801 IF A=B THEN 805
802 IF A>B THEN 806
803 IF A<B THEN 807
805 S$=" UNENTSCHIEDEN.":GOTO 810
806 S$=" FUER SIE! ":GOTO 810
807 S$=" FUER GEGNER! ":GOTO 810
810 PRINT:PRINT"SPIELSTAND:":PRINT
820 PRINT A;" ";B;S$
830 IF A=3 THEN 900
840 IF B=3 THEN 930
870 GOTO 300
900 REM GESAMTSIEG
910 PRINT:PRINT"SIE HABEN DAS SPIEL":PRINT
915 PRINT"MIT";A;" ";B;"GEWONNEN!"
920 GOSUB 1100
925 GOTO 950
930 PRINT:PRINT"SIE HABEN DAS SPIEL":PRINT
935 PRINT"MIT";A;" ";B;"VERLOREN!"
940 GOSUB 1200
945 GOTO 950
950 PRINT:PRINT
952 PRINT:PRINT:PRINT" NOCH EINMAL? J/N "
955 PRINT:PRINT" J ODER N EINGEBEN! "
960 GETA$:IFA$=""THEN 960
965 IFA$="J" THEN 980
970 IFA$="N" THEN 1300
975 GOTO 950
980 PRINT"Q"
985 A=0:B=0
990 GOTO 300
1000 REM PEEK IN
1010 P=PEEK(37136)
1020 P4=-(P AND 16)=16)
1030 P5=-(P AND 32)=32)
1040 P6=-(P AND 64)=64)
1050 P7=-(P AND 128)=128)
1060 RETURN
1100 REM MUSIC A
1110 POKE 36878,15
1115 FORL=1 TO 10
1120 FOR M=100 TO 235 STEP 2
1125 POKE 36875,M
1130 FOR N=1 TO 10
1135 NEXT N
1140 NEXT M
1145 POKE 36875,0
1150 FOR M=1 TO 100
1155 NEXT M
1160 NEXT L
1165 POKE 36878,0
1190 RETURN
1200 REM MUSIC B
1210 POKE 36878,15
1220 FOR M=240 TO 127 STEP -1
1230 POKE 36875,M
1240 FOR T=1 TO 80:NEXT
1250 NEXT M
1260 POKE 36878,0
1290 RETURN
1300 REM SCHLUSS
1310 PRINT"Q"
1320 PRINT" ICH BEDANKE MICH!":PRINT:PRINT
1330 PRINT" AUF WIEDERSPIELEN! "
1350 END

```

READY.

Listing 1. Programm für das
Knobelspiel zwischen zwei VC 20

den, wenn wechselweise der Programmteil Einlesen beziehungsweise Ausgeben gewählt wird. Es kann jedoch auch (nach POKE 37138, X mit POKE 37136, X ausgegeben und mit PRINT PEEK (37136) im Direktmode gearbeitet werden.

Knobelspiel Schere/Stein/Papier

Das Spiel, das zwischen zwei Spielern über zwei miteinander verkoppelte Computer gespielt wird, soll neben seinem Unterhaltungswert den Datenaustausch zwischen zwei Computern zeigen. Weiterhin zeigt es das Arbeiten mit dem logischen Befehlen AND und OR.

Üblicherweise wird das Knobelspiel nach folgenden Regeln gespielt: Die beiden Spielpartner zeigen auf Kommando zwei ausgestreckte Finger, bedeutet Schere, oder die flache Hand (Papier), oder die geballte Faust (Stein). Es gewinnt Schere gegen Papier (die Schere schneidet das Papier), Stein gegen Schere (der Stein schleift die Schere) und Papier gegen Stein (das Papier hüllt den Stein ein). Damit sind Sieg, Unentschieden und Niederlagen möglich. Gesamtsieger ist, wer zuerst drei Einzelspiele gewonnen hat.

Die logische Struktur des Spiels zeigt Bild 5. Man erkennt einen recht großen Aufwand an UND-beziehungsweise ODER-Schaltungen. Dabei ist bei der gezeigten Schaltung noch nicht verhindert, daß ein Spieler gleichzeitig mehrere Eingaben wählt. Außerdem fehlen noch die Zählschaltungen und ein Taktgeber. Wegen des hohen Aufwandes an Logikbausteinen bietet sich eine Computerschaltung mit entsprechendem Programm an. Die Grundstruktur mit den logischen Befehlen statt den UND-Verknüpfungen sieht folgendermaßen aus:

```

610 IF A 1 = 1 AND B 1 = 1 THEN 710
(Schere A — Schere B)
620 IF A 2 = 1 AND B 2 = 1 THEN 710
(Stein A — Stein B)

```

```

640 IF A 1 = 1 AND B 3 = 1 THEN 740
(Schere A — Papier B)

```

```

670 IF A 1 = 1 AND B 2 = 1 THEN 770
(Schere A — Stein B)

```

```

710 REM UNENTSCHIEDEN : END
740 REM SIEG A : END
770 REM SIEG B: END

```

Die bei den statischen Verknüpfungsgliedern folgenden ODER-


```

10 PRINT "3" REM VORBEREITUNG
20 REM P0...P3 OUT
30 REM P4...P7 IN
40 POKE 37138,15:POKE 37136,0
50 PRINT "A...AUSGEBEN":PRINT:PRINT
60 PRINT "E...EINGEBEN"
70 GET A$:IF A$="" THEN 70
80 IF A$="A" THEN 200
90 IF A$="E" THEN 450
100 GOTO 50
110 REM AUSGABE
120 P0=0:P1=0:P2=0:P3=0:X=0
130 PRINT:PRINT:PRINT "NOCH EINMAL? J/N"
140 GET A$:IF A$="" THEN 140
150 IF A$="J" THEN 200
160 IF A$="N" THEN 180
170 GOTO 130
180 POKE 37136,0
190 END
200 PRINT "3"
210 POKE 37136,X
220 PRINT:PRINT
230 PRINT "Z X O V EINGEBEN!"
240 PRINT:PRINT
250 GET A$:IF A$="" THEN 250
260 IF A$="Z" THEN 310
270 IF A$="X" THEN 330
280 IF A$="O" THEN 350
290 IF A$="V" THEN 370
300 GOTO 230
310 P0=1:X=1
320 GOTO 400
330 P1=1:X=2
340 GOTO 400
350 P2=1:X=4
360 GOTO 400
370 P3=1:X=8
380 GOTO 400
400 PRINT "P0=";P0;"P1=";P1;"P2=";P2;"P3=";P3
410 POKE 37136,X
420 GOTO 110
450 REM EINGABE
460 PRINT "3" GOSUB 500
470 PRINT "P=PEEK(37136)=";P:PRINT
480 PRINT "P4=";P4;"P5=";P5;"P6=";P6;"P7=";P7
490 FOR T=1 TO 400:NEXT:GOTO 450:END
500 REM PEEK IN
510 P=PEEK(37136)
520 P4=((PAND 16)= 16)
530 P5=((PAND 32)= 32)
540 P6=((PAND 64)= 64)
550 P7=((PAND 128)= 128)
560 RETURN

```

READY.

Listing 2. Programm zum Ein-/Auslesen über den Userport am VC 20

SIE HATTEN SCHERE IHR GEGNER SCHERE ALSO UNENTSCIEDEN SPIELSTAND: 0 : 0 UNENTSCIEDEN.	SIE HATTEN SCHERE IHR GEGNER PAPIER SIE HABEN GEWONNEN! SPIELSTAND: 1 : 0 FUER SIE!	SIE HATTEN PAPIER IHR GEGNER SCHERE GEGNER HAT GEWONNEN! SPIELSTAND: 2 : 3 FUER GEGNER! SIE HABEN DAS SPIEL MIT 2 : 3 VERLOREN!
---	---	---

Bild 7. Hardcopy des Spielablaufs

vorgeschalteten Blöcken, nämlich einer Überprüfung der Hardware und einer Kurzbeschreibung des Spieles. Nach der Aufforderung, die o.k.-Taste zu drücken, geht es weiter, wenn beide Spielpartner dies getan haben.

Es folgt die Aufforderung, eine Taste, bedeutet Schere oder Stein oder Papier, zu wählen. Haben bei-

de Spieler dies getan, so überprüft jeder Computer, wer gewonnen hat beziehungsweise ob das Einzelspiel unentschieden endete. Zwischenergebnis und Zwischenstand werden angezeigt. Jeder Computer addiert für sich die Siegpunkte von Spieler A beziehungsweise B. Nach Erreichen von drei Siegpunkten endet das Programm je nachdem mit einer aufsteigenden beziehungsweise abfallenden Tonfolge. Auf Wunsch kann das Spiel (ohne die Einleitung) wiederholt werden. Drei Hardcopies aus dem Spielverlauf zeigt Bild 7. Das vollständige Programm zeigt Listing 1.

— Für die Eingabe o.k., Schere, Stein beziehungsweise Papier sind im Programm die Funktionstasten F1 ... F4 vorgesehen. Dabei ist zu beachten, daß diese Tasten nicht schon anderweitig belegt sind (zum Beispiel beim Programmierhilfemodul). Eine Umprogrammierung auf andere Tasten ist leicht möglich.

— Das Programm kann notfalls auch mit Hilfe der eingangs beschriebenen Ein-/Ausgabe-Platine auf richtigen Ablauf getestet werden.

— Das Knobelprogramm benötigt in der vorliegenden Form mindestens die 3-KByte-Erweiterung. Bei Verzicht auf die REM-Zeichen und Verwendung von Doppelbefehlen genügt die Grundausrüstung des VC 20.

(Albrecht Langenstein)

Schaltungen werden im Computerprogramm dadurch realisiert, daß von drei Stellen aus an die entsprechende Einsprungstelle gesprungen wird.

Der logische Befehl OR wird an anderer Stelle benötigt. Die Zeilen 510 GOSUB 1000: REM EINLESEN 520 IF B 1 = 1 OR B 2 = 1 OR B 3 = 1 THEN 540

530 GOTO 510 540 REM VERGLEICH erlauben ein Weitergehen im Programm, wenn der Spielpartner B eine beliebige Eingabe, Schere oder Stein oder Papier, gemacht hat.

Anmerkung: Für A 1 bis B 3 steht im Listing P 1 bis P 6 (nach Bild 4).

Vorausgesetzt wurde, daß für beide Computer (als A und B bezeichnet) das exakt gleiche Programm verwendet werden soll (also kein Master- beziehungsweise Slave-

Programm). Die jeweilige Ausgabe Schere/Stein/Papier soll als logische 1 über die Ausgänge P 1, P 2 und P 3 ausgegeben werden beziehungsweise vom entgegengesetzten Computer über die Eingänge P 4, P 5 und P 6 eingelesen werden. Um das Spiel synchron zu halten, wird das »o.k.«-Signal benutzt, das über P 0 ausgegeben beziehungsweise über P 7 eingelesen wird. Hat zum Beispiel Spielpartner A die o.k.-Taste betätigt, so läuft das Programm erst weiter, wenn auch B sein o.k. signalisiert hat. A kann jetzt Schere, Stein oder Papier wählen. Wieder muß gewartet werden, bis auch B seine Wahl getroffen hat. Der synchrone Verlauf des Spiels wird durch akustische Signale unterstützt. Das vollständige Strukturdiagramm zeigt Bild 6.

Das Programm beginnt mit zwei

RUND UM DIE

Wenn man mit einem Computer arbeitet und Daten speichern will, steht man vor der Frage, »Welches Speichermedium ist für mich das geeignete?« Die verschiedenen Möglichkeiten der Speicherung auf Magnetband haben wir für Sie zusammengestellt.



Bevor wir die einzelnen Alternativen betrachten, sollten wir uns kurz mit den Grundlagen der Speicherung auf Band befassen. Die Übertragungsgeschwindigkeit der Daten wird in Baud angegeben. Ein Baud bedeutet die Übertragung eines Bits pro Sekunde. Die Daten werden durch bestimmte Frequenzen übertragen, die der Computer erkennen und interpretieren kann. Dabei muß ein bestimmter Takt eingehalten werden, damit die einzelnen Bits als solche erkannt werden können. Kommt es im Bandlaufwerk zu Geschwindigkeitsschwankungen, so können die Bits nicht richtig interpretiert werden, und es kommt zu Ladefehlern. Daher ist bei der Beurteilung von Bandlaufwerken die Laufgenauigkeit der Antriebsmotore ein entscheidender Faktor, den man nicht vernachlässigen darf.

Für die Commodore-Computer wird vom Hersteller selbst ein Bandlaufwerk mit dem Namen Datasette angeboten (Bild 1, links).

Die gute, alte Datasette

Mit dem Preis von 155 Mark ist die Datasette der billigste, von Commodore angebotene Datenspeicher. So werden sich viele Kunden bei dem Kauf ihres Computers durch diesen Aspekt haben leiten lassen, da man als Neuling auf dem Computermarkt meist Angaben über Übertragungsgeschwindigkeit gar nicht einordnen kann.

So kommt das böse Erwachen erst, wenn man größere Programme laden oder speichern will. Man sitzt vor seinem Bildschirm und wartet, und wartet, und wartet...

Bei dem Programm »Zauberschloß«, (siehe Happy-Computer, Ausgabe 2/84), dessen Volumen zirka 18 KByte umfaßt, beträgt die Ladezeit über sechs Minuten. Da die Datasette über ihren Anschluß verhältnismäßig viel Strom benötigt, kann es, wenn andere Geräte mit angeschlossen sind, zu Ladefehlern kommen, und man muß den gesamten Ladevorgang wiederholen. Dies ist besonders bei längeren Programmen sehr ärgerlich.

Trotz dieser Mängel ist die Datasette eine preiswerte Alternative gegenüber den sehr viel teureren Floppylaufwerken. Man muß bei der Arbeit mit der Datasette nur sehr viel Zeit mitbringen.

Der Markt für Interfaces, mit denen man seinen eigenen Recorder an den Computer anschließen kann,

DATASETTE



Bild 1. Die Data-
sette, der Data-
Kassettenrecorder
und ein Eigenbau-
Interface für han-
delsüblichen Kas-
settenrekorder

wächst ständig. Wir haben aus dem Angebot zwei Interfaces ausgewählt, um auch diese Möglichkeit vorzustellen. Das erste ist das Recorder Interface von Jeschke (Bild 2). Dieses Interface wird für 49 Mark angeboten. Als wir dieses Interface angefordert haben, wurde uns mitgeteilt, nicht jeder Recorder wäre damit anzusteuern. In unserem Testbetrieb (hauptsächlich japanische Produkte) haben wir allerdings keinen Recorder gefunden, der die übertragenen Daten nicht akzeptiert hätte.

Der eigene Recorder

Bei der Handhabung muß man allerdings einiges beachten. So muß man das Interface erst an den eigenen Recorder anpassen, um die Motorsteuerung ausnutzen zu können. Eine ausführliche Anleitung, wie die Verbindungskabel angeschlossen werden müssen, liegt der Beschreibung bei.

In der Produktvorschau dieser Firma wird für Ende Mai ein Steckmodul angekündigt, das mit dem Interface zusammen bis zu zehn mal schnellere Kassettenroutinen ermöglichen soll.

Die Heimwerkererlösung

Im Gegensatz zu anderen Heimcomputern wird beim C 64 der Kassettenrecorder an einen speziellen Kassettenport angeschlossen. Zum Anschluß eines Recorders wird daher ein spezieller Stecker benötigt. Dies ist schon der wesentlichste Unterschied zwischen dem Recorder von Commodore und normalen Recorders. Der Computer erkennt die Daten, die vom Recorder eingelesen werden nur dann, wenn sie in Form von Rechteckimpulsen am Port anliegen.

Beim »Schreiben« der Daten auf Kassette werden zwar Rechteckimpulse aufgenommen, durch das Band und den Aufnahme-/Wieder-

gabevorgang jedoch werden diese Rechtecke zu einem sinusförmigen Signal verformt. Diese Daten kann der Computer nicht mehr richtig interpretieren. Um dies zu vermeiden, ist eine kleine Schaltung erforderlich, die bei der Datasette bereits eingebaut ist.

Ferner besitzt diese einen kleinen Schalter, der beim Drücken einer Taste geschlossen wird. Daher kann der Computer feststellen, ob eine Taste gedrückt wurde.

Mit etwas Geschick und Material für zirka 15 Mark kann auf einfache Weise ein Anschlußkabel gebaut werden.

Die Schaltung

Bild 3 zeigt die Belegung der Kontakte am Kassettenport und die Schaltung. Im Baustein SN 7414 sind sechs invertierende Schmitt-Trigger enthalten, von denen zwei benötigt werden. Ein Schmitt-Trigger setzt (vereinfacht ausgedrückt) ein Sinus-signal in ein Rechtecksignal um.

Da wir invertierende Schmitt-Trigger haben, müssen wir zwei verwenden, damit die Polarität wieder stimmt. Das Wiedergabesignal vom Recorder läuft nun über die beiden Schmitt-Trigger und erhält so wieder seine ursprüngliche Form. Die Spannungsversorgung (+5 Volt) kann direkt am Kassettenport abgenommen werden.

Die Schreibleitung, die Leseleitung und die Leitung »Kassettenmotor« vom Computer werden an Klinkenstecker auf der einen Seite und an einen Platinenstecker auf der anderen Seite angelötet.

Der IC 7414 kann in den Platinenstecker eingebaut werden. Dazu sind die nicht verwendeten Anschlußbeine des IC und die nicht verwendeten Lötösen des Steckers abzuschneiden. Zunächst wird Pin 14 vom IC an Stift 2 des Steckers angelötet und das IC senkrecht gestellt (Bild 4). Pin 2 und Pin 3 werden direkt am IC verlötet. Von Pin 4 wird eine Drahtbrücke nach Stift 4 gelötet, ebenso von Pin 7

nach Stift 1 und Stift 6. An Stift 1 kommt die Masseleitung, an Stift 3 die Leitung für den Fernbedienungsstecker (REMOTE), an Stift 5 die Leitung für den Mikrofoneingang. Am Pin 1 des IC wird die Leitung vom Ohrhörerausgang angelötet. Nun können die Klinkenstecker angelötet und der Platinenstecker in sein Gehäuse eingebaut werden.

Der erwähnte Schalter im Commodore-Recorder wird hier durch eine Drahtbrücke (Stift 6 und 1) im Platinenstecker fest auf »eingeschaltet« verdrahtet. Wenn der C 64 zum Drücken einer Taste auffordert, so ist diese für den Computer bereits gedrückt und der Schreib-/Lesevorgang beginnt sofort. Daher sollten die entsprechenden Recordertasten schon gedrückt sein, bevor zum Beispiel ein Programm geladen wird. Es kann auch ein kleiner Schalter an Stelle der Drahtbrücke eingebaut werden.

Bild 5. Das Minikassettengerät FS-9600

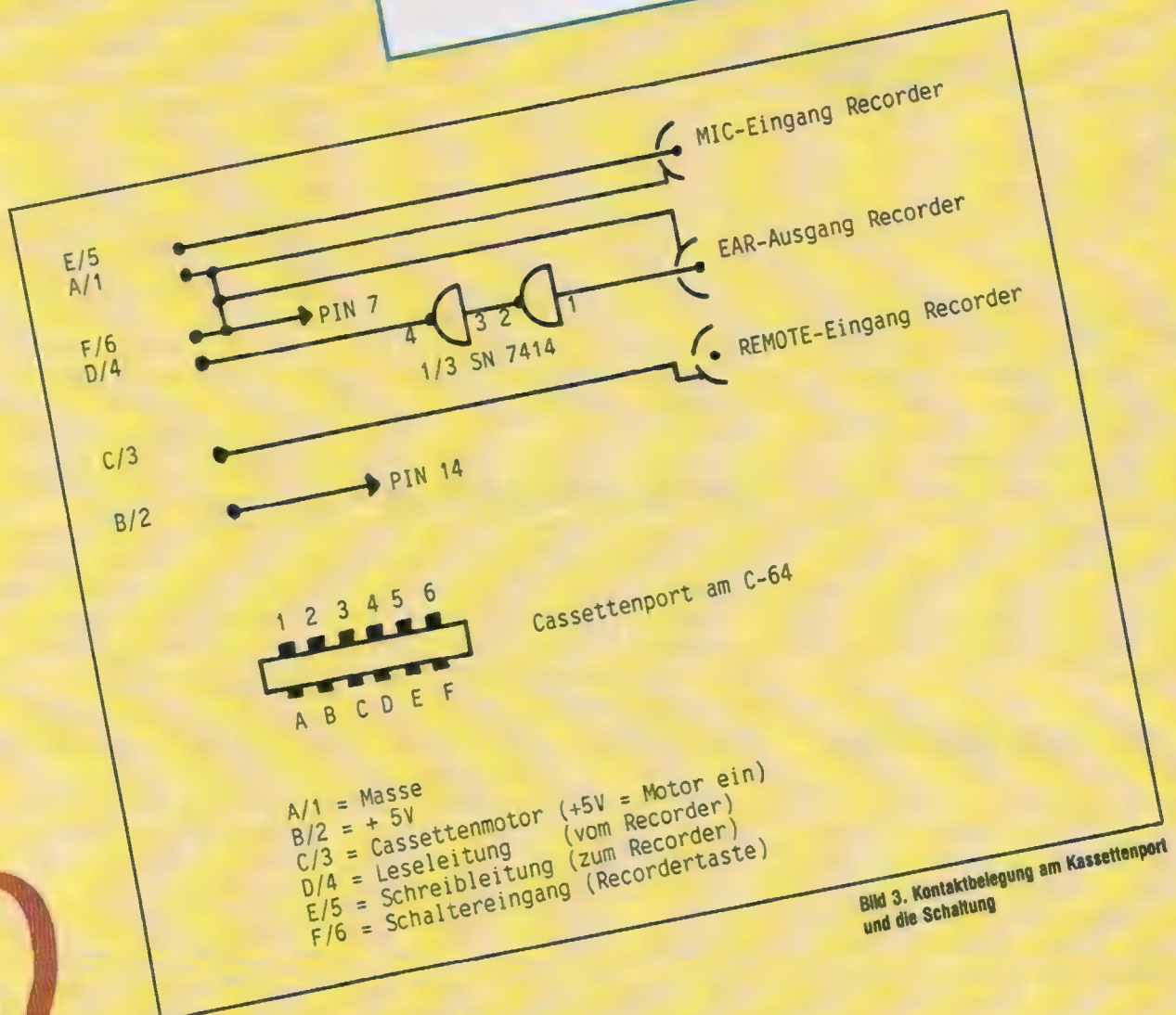


Bild 3. Kontaktbelegung am Kassettenport und die Schaltung

Da der Recorder seine Spannung vom Computer bekommt, sollten die Batterien herausgenommen oder der Netzstecker des Recorders abgezogen werden. Nach dem Schreib-/Lesevorgang wird die Stromversorgung für den Kassettensmotor erst dann wieder freigegeben, wenn der Computer erkennen konnte, daß am Recorder keine Taste mehr gedrückt ist. Da aber statt des Schalters eine Drahtbrücke eingebaut ist, kann der Recorder nur noch dann eingeschaltet werden, wenn der Computer es «will». Das kann beim Umspulen der Kassette unerwünscht sein. Wenn der Platinenstecker gezogen und wieder gesteckt wird, ist das Problem behoben. Der Recorder sollte dabei ausgeschaltet sein.

Der Zusammenbau ist für einen Bastler recht einfach. Wer jedoch keine Erfahrung im Umgang mit dem Lötkolben hat, sollte den Aufbau lieber einem Freund überlassen. Das Interfacekabel funktioniert ohne Probleme auch bei den übrigen Commodore-Computern. Die Bauteile sind im einschlägigen Fachhandel erhältlich.

Das Interface (Bild 1, Mitte, Chinchstecker und Interface) funktioniert nur mit solchen Recordern, die mit einer Spannung von 4 bis 6 Volt betrieben werden können und bei denen die Plusleitung nicht an Masse liegt.

Die meisten Recorder erfüllen zwar diese Voraussetzung, doch sollte man sich sicherheitshalber vergewissern. Testen kann man dies ohne Gefahr für den Recorder folgendermaßen: In den REMOTE-Eingang des Recorders einen Klinkestecker einstecken. Wenn der Recorder nun eingeschaltet wird, so läuft er nicht.

Dann die Batterien aus dem Recorder nehmen (oder Netzstecker ziehen) und eine 4,5 Volt Flachbatterie mit dem Pluspol am Außenstift des REMOTE-Steckers und dem Minuspol am Außenstift des MIC-Steckers anschließen. Der Recorder muß nun wieder laufen und die Kassette sollte in der richtigen Richtung abgespielt werden. Wenn das nicht der Fall ist, dann ist dieser Recorder nicht geeignet.

Das Substitutprodukt

Der Data Kassettenrecorder von Nettetaler Modell PM-4401 (Bild 1, rechts) tritt als direkter Konkurrent der Original Datasette auf. Der Preis dieses Gerätes liegt bei 109 Mark.

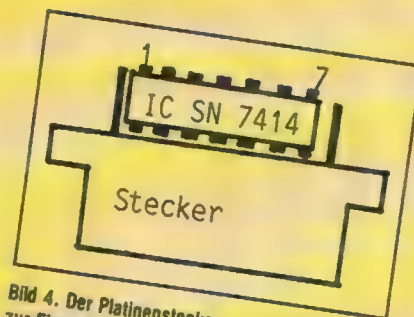


Bild 4. Der Platinenstecker zur Eigenbaulösung

Leider hatte unser Testgerät einen Fehler, und das neue Testgerät hat uns vor Redaktionsschluß nicht mehr erreicht. Ein Test wird nachgeholt.

Der Superschnelle

Die Übertragungsraten der bisher vorgestellten Produkte sind sehr bescheiden. Will man mit ihnen arbeiten, muß man viel Zeit mitbringen. Dafür sind sie aber auch recht preiswert.

Den stolzen Preis von 980 Mark muß man für das Minikassettenge-

lich gemacht. So brauchte die Datasette über sechs Minuten, um dieses Programm zu laden. Das Floppy-Laufwerk 1541 benötigt zirka 51 Sekunden. Lädt man das Programm Zauberschloß mit dem Minikassettenlaufwerk, muß man nur noch zirka 28 Sekunden warten.

Würde nur dieser Zeitvorteil bestehen, wäre der Preis sicher nicht gerechtfertigt. Bei der FS-9600 handelt es sich jedoch um ein »intelligentes« Gerät. So hat man hier alle Vorzüge, die sonst nur mit einem Floppy-Laufwerk realisiert werden können. Bei der Arbeit mit dem Minikassettengerät steht zum Beispiel ein Inhaltsverzeichnis der Kassette zur Verfügung, man kann Blockweise editieren und es gibt eine »APPEND« Funktion, mit der man Programme direkt auf der Kassette verketteten kann.

Positiv ist auch zu werten, daß bei auftretenden Fehlern eine Statusvariable zur Verfügung steht, aus der man anhand einer Fehlerliste den auftretenden Fehler lokalisieren und beseitigen kann.

Das Resultat: von 15 bis 980 Mark

Wie wir gesehen haben, gibt es viele Möglichkeiten, Daten auf Band zu speichern. Finanziell reicht die Palette von zirka 15 Mark für die Heimwerkerlösung bis zu 980 Mark für die schnelle Minikassette FS-9600. Für welche Möglichkeit man sich entscheidet, bleibt wohl von der Anwendung und vor allem vom Geldbeutel abhängig.

(M.Sandweg/rg)



Bild 2. Das käufliche Interface für den eigenen Audio-Recorder

rät FS-9600 (Bild 5) von Lothar Schanuel ausgeben. Für 680 Mark ist die Bausatz-Version dieses Gerätes zu bekommen.

Die Schnelligkeit, die dieses Gerät an den Tag legt, ist mehr als verblüffend. Mit einer Übertragungsgeschwindigkeit von 9000 Baud stellt die FS-9600 sogar das Floppy-Laufwerk 1541 von Commodore in den Schatten. Wenn wir, wie schon bei der Datasette, wieder das Programm Zauberschloß heranziehen, werden die Zeitunterschiede deut-

EPSON FÜR COMMODORE 64

Ein Beweis für die Vielseitigkeit des User-Ports ist seine Programmierung als Centronics-Schnittstelle.

Daß dies mit wenig Material, einem Lötkolben und einer Stunde Zeit zu realisieren ist, zeigt dieser Bericht am Beispiel der Epson-Drucker.

Für den Anschluß eines Druckers an den C 64 ist zunächst nur der serielle Bus, an dem auch das Diskettenlaufwerk angeschlossen ist, vorgesehen. Dieses Konzept, mehrere Geräte an einer Schnittstelle zu betreiben, hält zum einen die Anzahl der Anschlüsse am Computer in Grenzen, zum anderen genügt für die Verbindung ein einfaches Kabel. Des weiteren kann man sich auf ein einheitliches Protokoll zur Datenübertragung von und zu den Geräten beschränken (bei Commodore mittels OPEN, CLOSE, CMD, PRINT mit logischer Datei- und Gerätenummer).

Der Nachteil dabei ist, daß man nur die von Commodore angebotene Peripherie direkt anschließen kann, wie zum Beispiel die Drucker 1526, 1525, 1520 oder den MPS 801. Möchte man aber einen Drucker eines anderen Herstellers wie zum Beispiel einen Epson MX/RX und FX-80 (Bild 1) anschließen, ist es mit einer einfachen Kabelverbindung oft nicht mehr getan. Standard bei Druckern ist nicht der serielle Bus von Commodore, sondern die Centronics- und die RS232-Schnittstelle. Die Epson-Drucker sind dabei in der Regel mit einer Centronics-Schnittstelle ausgestattet.

Bild 2.
Das Verbindungskabel vom User-Port zur Centronics-Schnittstelle



Die Lösung dieses Anschlußproblems ist auf zwei verschiedenen Wegen möglich. Entweder wird ein vergleichsweise teures Interface zwischen C 64 und dem Epson zwischengeschaltet beziehungsweise in den Drucker eingebaut, oder, was preiswerter und flexibler ist, es wird der User-Port als Centronics-Schnittstelle programmiert.

Voraussetzung dafür ist allerdings, daß der User-Port nicht anderweitig benötigt wird (EPROM-Brenner, Modem, RS232 etc.). Von der elektronischen Seite gibt es gegen eine Direktverbindung keine

Einwände, solange der Eingang am Drucker den mit dem User-Port verbundenen CIA nicht mehr als mit den im Datenblatt spezifizierten 3,2 mA im Low-Zustand pro Signalleitung belastet. Wir haben an einem Epson-FX-80-Drucker einen Eingangs-Kurzschlußstrom von 1,6 mA gemessen, womit ein sicherer Betrieb des CIA garantiert ist. Außer dem beschriebenen Kabel ist keine weitere Hardware erforderlich.

Man benötigt nun einen 24poligen User-Port-Stecker (zum Beispiel TRW 251-12-50-17081/350-4SN-9) sowie einen 36poligen Centronics-Stecker (zum Beispiel Amphenol



Bild 1. Der Epson-FX-80-Drucker

Bild 3. Der Verkabelungsplan

Drucker		User-Port	
Pin Nummer	Bezeichnung	Bezeichnung	Pin Nummer
1	STROBE	PA2	M
2	DATA 1	PB0	C
3	DATA 2	PB1	D
4	DATA 3	PB2	E
5	DATA 4	PB3	F
6	DATA 5	PB4	H
7	DATA 6	PB5	J
8	DATA 7	PB6	K
9	DATA 8	PB7	L
10	ACKNLG	FLAG2	B
16	GND	GND	A

57-30360). Die Centronics-Konvention verlangt ein abgeschirmtes Kabel und schreibt vor, daß alle Signalleitungen mit einer eigenen Signalmasse-Leitung verdrillt werden müssen. Unser C 64 arbeitet aber trotz Verzicht auf diesen Aufwand bisher fehlerfrei mit einem FX-80 zusammen.

Das Kabel (Bild 2 und 3): Es müssen elf Verbindungen hergestellt werden (Flach- und Rundkabel).

Die Treibersoftware

Die Software arbeitet einwandfrei mit Programmen zusammen, die eine ASCII-Ausgabe-Option besitzen, wie dies bei den meisten Textverarbeitungsprogrammen der Fall ist. Es soll hier auch lediglich der Hard- und Software-Kern der Schnittstelle dargestellt werden.

Das Programm ist mit 87 Byte Länge sehr kurz und kann damit im

Kassetten-Puffer (\$033C bis \$03FB), oder wie hier im sonst ungenutzten Bereich (\$02A7 bis \$02FF) untergebracht werden. Es wirkt wie folgt: Die Standard-Zeichenausgaberroutine des Betriebssystems, CHROUT, wird durch einen geänderten Vektor in das Erweiterungsprogramm umgeleitet. Dieses überprüft, ob die Ausgabe über Gerätenummer 4 (Standardnummer für Drucker) erfolgen soll. Wenn nicht, springt es sofort in das Betriebssystem zurück. Ansonsten wird das zu übertragende Byte an CIA2 Port B gelegt und über einen Strobe-Impuls über Port A, Bit 2 dem Drucker mitgeteilt, daß die anliegenden Daten gültig sind. Der Drucker quittiert den Dateneingang über das Acknowledge-Signal. Dieses setzt am C 64 über den FLAG-2-Eingang Bit 4 des Interrupt-Registers von CIA 2. Der CIA wird so programmiert, daß dadurch aber kein Interrupt ausgelöst wird. Das Programm wartet viel-

mehr, bis dieses Bit gesetzt ist, löscht es beim Auslesen automatisch und springt erst dann in das rufende Hauptprogramm zurück. Damit ist ein voller Handshake gewährleistet.

Aus dem Assemblerlisting (Bild 4) geht eine Gliederung in drei Abschnitte hervor:

1. Zeile 190-280

Abfrage, ob Ausgabe über Gerätenummer 4 erfolgen soll

Bei ASCII-Code 13 (Carriage Return) wird automatisch noch ein ASCII-Code 10 (Line Feed) gesendet. Durch Weglassen der Zeilen 250 bis 280 wird kein automatischer Line Feed gesendet.

2. Zeile 320-420

Ausgabe eines Byte

3. Zeile 500-670

Programmierung des CIA2 für die Ausgabe. Umstellen des Ausgabevektors auf den Programmteil 1

Die praktischen Anwendungen

Der Umgang mit dieser Schnittstelle gestaltet sich im praktischen Betrieb recht einfach und problemlos. Das Programm wird in der vorliegenden Form in den Bereich \$02A7 bis \$02FD geladen und durch Aufruf des Initialisierungsteils (JSR \$02D4 beziehungsweise SYS 724) für die Druckausgabe vorbereitet. Das kann zum Beispiel der angefügte Basic-Lader (Bild 5) übernehmen. Man kann dann zum Beispiel ein Basic-Programm mit folgender Sequenz listen:

OPEN 1,4 : CMD 1 : LIST

Damit werden sämtliche Daten über den Drucker ausgegeben und nach:

PRINT # 1 : CLOSE 1

werden die Daten wieder auf den Bildschirm ausgegeben.

Man beachte, daß nach Drücken der Kombination RUN/STOP RESTORE (non maskable Interrupt) sämtliche Zeiger auf ihre Standardwerte zurückgestellt sind. Deshalb muß die Schnittstelle durch SYS 724 wieder neu initialisiert werden.

Die Möglichkeiten, einen Drucker wie den Epson MX/FX-80 mit einer solchen Schnittstelle zu betreiben, sind beinahe unbegrenzt. So ist es beispielsweise möglich, ein Hardcopyprogramm zu schreiben, mit dem Ausdrücke in der Form des Bildes 6 ausgedruckt werden können. Ein Epson FX-80 braucht für ein solches Bild etwa 32, in doppelter Größe zirka 70 Sekunden. Auch ist es möglich, im Grafik-Modus einen

USER PORT ALS CENTRONICS-SCHNITTSTELLE

(C) 1983 PROF1-ASS 64

```

2
140: 02A7
150: 02A7

.OPT P
*= $2A7

; EINSPRUNGSTELLE FUER KERNAL CHROUT
; AUSGABE EINES ZEICHENS MIT AUTO-LINE FEED

CHROUT PHA $9A ; NUMMER DES AUSGABEBERAETS
LDA $4 ; DRUCKER "?"
CMP $4
BEQ DR
JMP $F1CD ; ZURUECK ZUM KERNAL

PLA
CMP $13 ; CR "?"
BNE CENT ; BYTE AUSGEBEN
JSR CENT ; ZUERST CR AUSGEBEN
LDA $10 ; DANN NOCH LF

; EIN BYTE AN CENTRONICS-SCHNITTSTELLE AUSGEBEN

320: 02BB 8D 01 DD CENT
330: 02BE AD 00 DD
340: 02C1 29 FB
350: 02C3 8D 00 DD
360: 02C6 09 04
370: 02C8 8D 00 DD
380: 02CB AD 0D DD WARTEN
390: 02CE 29 10
400: 02D0 F0 F9
410: 02D2 18
420: 02D3 60

STA $DD01 ; PORT #
LDA $DD00 ; PORT #
AND $FB ; PA2 = 0 STROBE-IMPULS SETZEN
STA $DD00 ; PA2 = 1 STROBE-IMPULS RUECKSETZEN
ORA $04
STA $DD00 ; INTERRUPT DATA
LDA $DD0D ; BIT 4 = FLAG 2
AND $10 ; WARTEN BIS ES GESETZT IST
BEQ WARTEN
CLC
RTS

; INITIALISIERUNG DER CENTRONICS-SCHNITTSTELLE

; STROBE PA2
; ACKNLG FLAG 2
; BUSY WIRD NICHT AUSGEWERTET

INIT SEI
LDA $FF ; PORT # AUSGANG
STA $DD03 ; DDRA
LDA $DD02 ; PA2 AUSGANG
ORA $04
STA $DD02 ; PORT A
LDA $DD00 ; PA2 (STROBE) = 1
ORA $04
STA $DD00
LDA $10
STA $DD0D ; LOESCHE INTMASK BIT #
LDA $DD0D ; LOESCHE INT DATA
LDA $<CHROUT
STA $326 ; AUSGABEVEKTOR UMSTELLEN
LDA $>CHROUT
STA $327
CLI
RTS

```

500: 02D4 78
510: 02D5 A9 FF
520: 02D7 8D 03 DD
530: 02DA AD 02 DD
540: 02DD 09 04
550: 02DF 8D 02 DD
560: 02E2 AD 00 DD
570: 02E5 09 04
580: 02E7 8D 00 DD
590: 02EA A9 10
600: 02EC 8D 0D DD
610: 02EF 8D 0D DD
620: 02F2 A9 A7
630: 02F4 8D 26 03
640: 02F7 A9 02
650: 02F9 8D 27 03
660: 02FC 58
670: 02FD 60
U02A7-02FE
READY.

neuen Zeichensatz mit near-Letter-Quality zu entwerfen. Dieser druckt dann, zwar deutlich langsamer, aber dafür wesentlich schöner, zum Beispiel wichtige Briefe, aus.

Die Nachteile dieser Lösung sind natürlich immer im Verhältnis zum finanziellen Aufwand zu sehen. So ist es mit dem abgebildeten Basicprogramm nicht möglich, die Steuer- und Grafikzeichen des Commodore 64 auszudrucken. Auch der gezeigte Hardcopyausdruck ist nur mit einem zusätzlichen Programm zu realisieren.

Bei weiterem Interesse können Sie sich an die Autoren wenden.

(A. Wängler/T. Krätzig)

Bild 4. Das Assemblerlisting

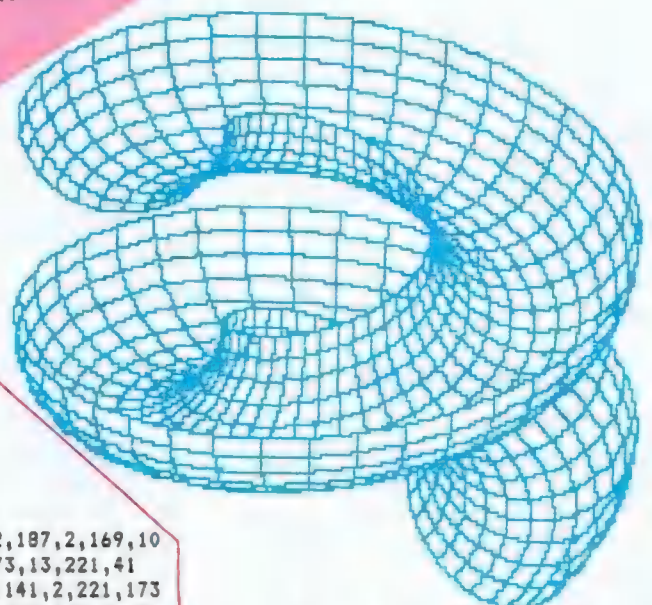


Bild 6. Probeausdruck einer Hardcopy

```

100 PRINT"LOAD CENTRONICS-SCHNITTSTELLE WAIT"
105 PRINT:PRINT" MIT AUTO-LINE-FEED"
110 PRINT" THOMAS KRAETZIG 21.2.84"
120 PRINT" BITTE GEEIGNETES KABEL VERWENDEN"
130 RESTORE
140 FOR I=0 TO 86
150 READ X:POKE 679+I,X
160 NEXT
180 SYS 724
190 END
191 DATA 72,165,154,201,4,240,3,76,205,241,104,201,13,208,5,32,187,2,169,10
192 DATA 141,1,221,173,0,221,41,251,141,0,221,9,4,141,0,221,173,13,221,41
193 DATA 16,240,249,24,96,120,169,255,141,3,221,173,2,221,9,4,141,2,221,173
194 DATA 0,221,9,4,141,0,221,169,16,141,13,221,173,13,221,169,167,141,38,3
195 DATA 169,2,141,39,3,88,96

```

READY.

Bild 5. Der Basic-Lader

Strukturiertes Programmieren

Nachdem wir in der letzten Ausgabe ein einfaches Spiel entwickelt haben, das zum Teil mit Struktogrammen dokumentiert wurde, bringen wir jetzt den Rest des Listings und eine Gegenüberstellung von Flußdiagramm und Struktogramm.

Diese Ausführung des Spiels »TIC TAC TOE« ist recht einfach. Nach gründlicher Überlegung könnte man dieses Programm sofort kodieren. In der Praxis und mit wachsender Programmiererfahrung werden die Aufgaben jedoch bald umfangreicher sein. Professionelle Programmierer fertigen sich, bevor sie anfangen zu kodieren, eine grafische Darstellung des Programmlösungsweges an. Grafiken überblickt der Mensch wesentlich schneller als einfachen Text oder sogar Zahlenkolonnen. Das wohl am häufigsten eingesetzte grafische Hilfsmittel ist der Programmablaufplan (auch Flußdiagramm, Blockdia-

gramm oder im Englischen: Program Flowchart genannt). Es wird jedoch in den letzten Jahren immer mehr von Struktogrammen (auch Nassi-Shneidermann-Diagramm oder NS-Diagramm genannt) abgelöst. Der Hauptgrund war wohl die Entwicklung höherer Programmiersprachen, die eine Strukturierung eines Programms zulassen. Vor allem sind das Pascal, Fortran 77, Cobol, und PL/I. Selbst die neuesten Basic Versionen (zum Beispiel das extended Microsoft-Basic oder, etwas eingeschränkt, das Basic 3.5 der neuen Commodore 16 und 264) besitzen Befehle, die eine Strukturierung eines Programms zulassen. Da-

mit ist vor allem der vollkommene Verzicht von GOTO-Sprunganweisungen gemeint. Im Gegensatz zu Programmablaufplänen (PAP) ist in Struktogrammen eine Darstellung von GOTO-Befehlen nicht vorgesehen und wird somit auch nicht unterstützt. Wer mit normalem Standard-Basic arbeitet, das keine Strukturierungsbefehle wie IF..THEN..ELSE, REPEAT UNTIL, DO WHILE und so weiter kennt, braucht dennoch nicht auf Struktogramme verzichten. Wenn man sich an die im Heft 4/84 beschriebenen Regeln zur Strukturierung hält, ist eine Anwendung von Struktogrammen sogar recht sinnvoll.

Fortsetzung auf Seite 40

```

10 REM *****
20 REM * KREUZ UND QUER *
30 REM *****
40 PRINT "L"
100 DIM SS(9):DIM BW(8)
110 GOSUB1000:REM SPIELFELD INITIALISIEREN
120 GOSUB2000:REM ANZEIGEN
130 GOSUB3000:REM ZUG HOLEN
140 GOSUB2000:REM ANZEIGEN
150 GOSUB4000:REM AUF SPIELLENDE PRUEFEN
160 GOSUB5000:REM COMPUTERZUG
170 GOSUB2000:REM ANZEIGEN
180 GOSUB4000:REM AUF SPIELLENDE PRUEFEN
190 GOTO130
1000 REM-----
1010 REM INITIALISIERUNG
1020 :
1100 FORP=1TO9
1110 :SS(P)=0
1120 NEXTP
1130 RETURN
2000 REM-----
2010 REM ANZEIGEROUTINE
2020 :
2100 FORP=1TO9
2110 :IFSS(P)=1THENPRINT "X";
2120 :IFSS(P)=-1THENPRINT "O";
2130 :IFSS(P)=0THENPRINT ". ";
2140 :IFINT(P/3)=P/3THENPRINT
2150 NEXTP
2160 PRINT:PRINT
2170 RETURN
3000 REM-----
3010 REM SPIELERZUG
3020 :
3040 INPUT "GIB ZUG EIN ";P
3050 IFSS(P)<>0THENPRINT "DIESES QUADRAT
IST BESETZT":GOTO3000
3060 SS(P)=1
3070 RETURN
3090 :

```

```

4000 REM-----
4010 REM AUF SPIELLENDE PRUEFEN
4020 :
4040 GOSUB6000
4050 FORP=1TO8
4060 :IFBW(P)=-3THENPRINT "ICH GEWINNE":E
ND
4070 :IFBW(P)=3THENPRINT "DU GEWINNST":E
ND
4080 NEXTP
4090 FORP=1TO9
4100 :IFSS(P)=0THENRETURN
4110 NEXTP
4120 PRINT "UNENTSCHEIDEN":END
4130 :
5000 REM-----
5010 REM COMPUTERZUG
5020 :
5030 GOSUB6000
5040 MT=0
5050 FORP=1TO8
5060 :IFBW(P)=-2THENGOSUB7000:RETURN
5070 :IFBW(P)=2THENMT=MT+1
5080 NEXTP
5090 IFMT=2THENPRINT "AUCH GUT. ANGEBER!"
:END
5100 IFMT=1THENGOSUB8000:RETURN
5110 GOSUB9000
5120 RETURN
5130 :
6000 REM-----
6010 REM SPIELFELD BEWERTEN
6020 :
6030 :
6040 BW(1)=SS(1)+SS(2)+SS(3)
6050 BW(2)=SS(4)+SS(5)+SS(6)
6060 BW(3)=SS(7)+SS(8)+SS(9)
6070 BW(4)=SS(1)+SS(4)+SS(7)
6080 BW(5)=SS(2)+SS(5)+SS(8)
6090 BW(6)=SS(3)+SS(6)+SS(9)
6100 BW(7)=SS(1)+SS(5)+SS(9)
6110 BW(8)=SS(3)+SS(5)+SS(7)

```

```

6120 RETURN
6130 :
7000 REM-----
7010 REM SIEGZUG
7020 :
7040 GOSUB9000
7050 GOSUB6000
7060 FORP=1TO8
7070 :IFBW(P)=-3THENRETURN
7080 NEXTP
7090 GOSUB10000
7100 GOTO7000
7110 :
7120 :
8000 REM-----
8010 REM SPERREN
8020 :
8040 GOSUB9000
8050 GOSUB6000
8060 F=0
8070 FORP=1TO8
8080 :IFBW(P)=2THENF=1
8090 NEXTP
8100 IFF=0THENRETURN
8110 GOSUB10000
8120 GOTO8000
8130 :
9000 REM-----
9010 REM ZUFALLSZUG
9020 :
9040 CM=9*RND(0)+1
9050 IFSS(CM)<>0THEN9000
9060 SS(CM)=-1
9070 RETURN
9080 :
9090 :
10000 REM-----
10010 REM ZURUECKNEHMEN
10020 :
10040 SS(CM)=0
10050 RETURN

```

Listing. Das im letzten Heft entwickelte Spiel »TIC TAC TOE« als komplettes Listing. Verbessern Sie die Grafik und die Strategie.

Doch zunächst möchte ich Ihnen die Symbolik des PAP vorstellen. Schauen wir uns die Tabelle an. Hier sind die wichtigsten und gebräuchlichsten Elemente des PAP aufgeführt.

Symbol 1: Bearbeitung

Man trägt Anweisungen ein; die Schreibweise der Eintragung orientiert sich oft an der zu verwendenden Programmiersprache. Es kann auch eine Gruppe von Anweisungen eingetragen werden.

Symbol 2: Verzweigung

Wenn in einem Programm in Abhängigkeit einer Bedingung der lineare Ablauf unterbrochen werden soll, erfolgt die Eintragung der Bedingung in diesem Symbol.

Symbol 3: Unterprogrammaufruf

Hier wird die Bezeichnung eines Unterprogramms eingetragen. In Basic setzt man am Besten noch die Anfangszeilennummer dazu.

Symbol 4: Programmmodifikation

Hier trägt man eine Aktion ein, die den Ablauf an einer zeitlich dahinterliegenden Stelle ändert (wird selten gebraucht).

Symbol 5: Operation von Hand

In manchen Programmen kommt es vor, daß an einer bestimmten Stelle die Diskette oder Kassette gewechselt werden muß. Diese Aktion wird hier eingetragen.

Symbol 6: Eingabe, Ausgabe

Man benutzt dieses Symbol immer, wenn eine Eingabe- oder Ausgabeanweisung im Programmablauf auftritt. Man trägt dabei die Art der Ein-/Ausgabe ein (INPUT oder PRINT oder ähnliches) und den Datensatz, der gelesen oder geschrieben wird.

Symbol 7: Flußlinie

Sie verbindet die Symbole miteinander. Die Pfeilspitzen können weggelassen werden, wenn die Flußrichtung eindeutig ist (normalerweise immer von oben nach unten, von links nach rechts).

Symbol 8: Konnektor, Übergangsstelle

Die Eintragung ist beliebig. Dieses Symbol wird benutzt, um nicht durch zu viele Flußlinien die Übersicht zu verlieren.

Symbol 9: Grenzstelle, Anschlußmarke

Dieses Symbol enthält eine Bezeichnung von Programmbeginn und -ende sowohl von Haupt- als auch von Unterprogrammen.

Symbol 10: Bemerkung

Falls der Platz innerhalb eines Symbols nicht ausreicht, um es zu beschriften, kann man mit diesem Symbol eine Bemerkung hinzufügen.

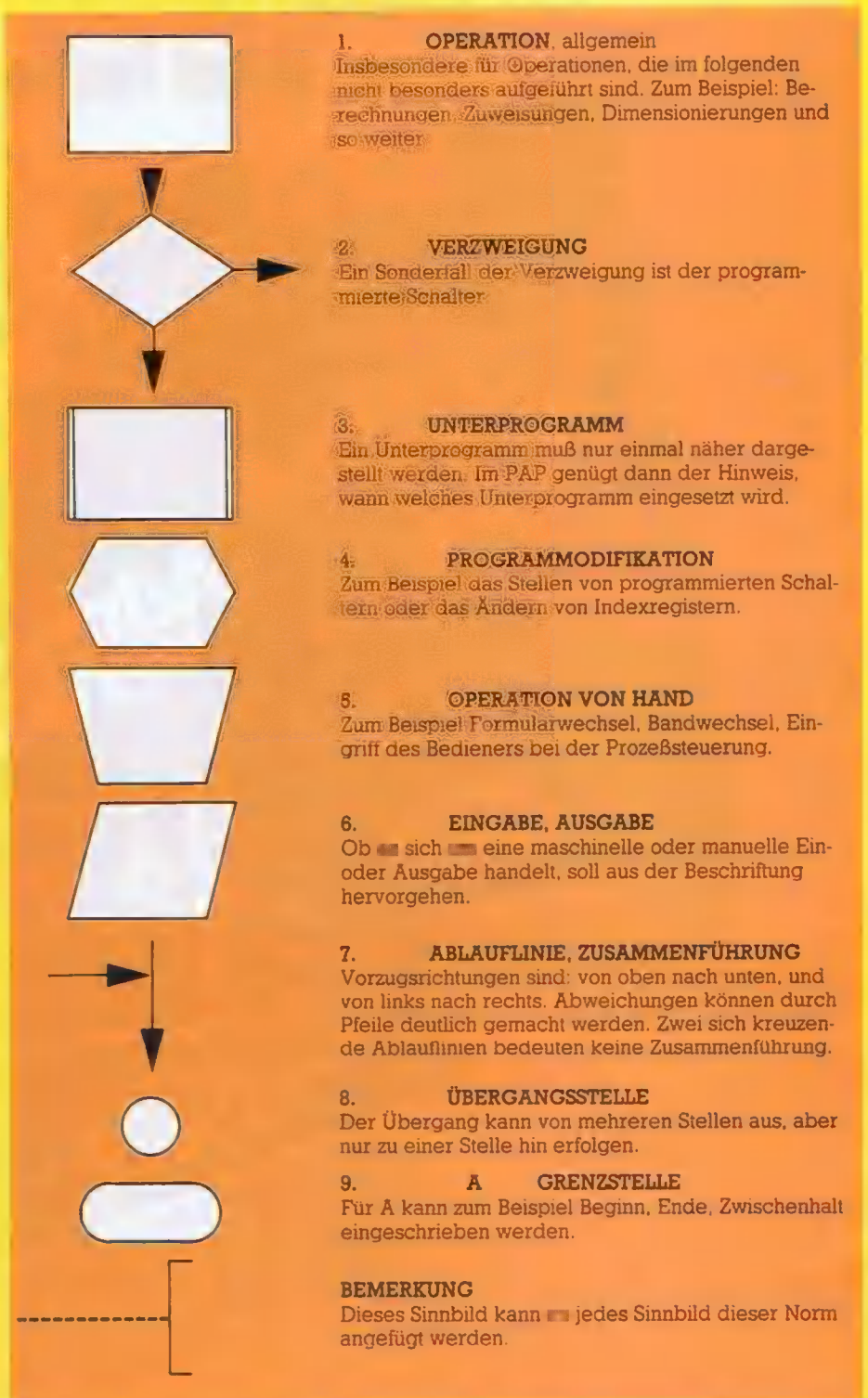


Tabelle 1: Sinnbilder für Programmablaufpläne

Unternehmen gehen bei Programmiervorhaben immer mehr dazu über, anstatt von Flußdiagrammen Struktogramme einzusetzen. Sie zwingen den Programmierer zu einer übersichtlichen Programmierung und erlauben es ihm nicht, durch viele GOTOs das Programm »undurchsichtig« zu machen. Wenn man Struktogramme verwendet, kann man auf Programmablaufpläne völlig verzichten. Als Programmierer sollte man beide Darstellungsarten kennen, denn Programmablaufpläne und Strukto-

gramme werden noch lange nebeneinander bestehen.

Die Elemente des Struktogramms

Jede Aufgabenstellung läßt sich mit den folgenden Grundstrukturen lösen:

Sequenz Auswahl Wiederholung

Diese Grundstrukturen werden zu Strukturblocken zusammengesetzt und sind beliebig ineinander verschachtelbar.

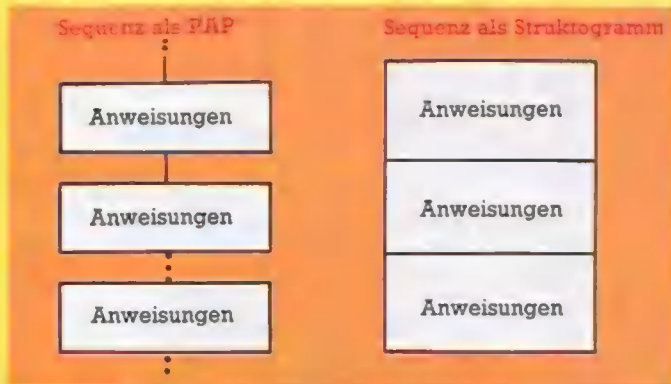


Bild 1. Die Sequenz ▲

Bild 3. Die Anweisungen 1 werden nur dann ausgeführt, wenn die Bedingung B erfüllt ist, sonst werden die Anweisungen 2 ausgeführt.

Bild 6. Die Anweisungen werden ausgeführt, und wenn die Wiederholungsbedingung erfüllt ist, werden die Anweisungen wieder ausgeführt, sonst wird die Schleife verlassen. Um eine Endlosschleife zu vermeiden, muß auch hier die Wiederholungsbedingung innerhalb der Gruppe der Anweisungen verändert werden. ▼

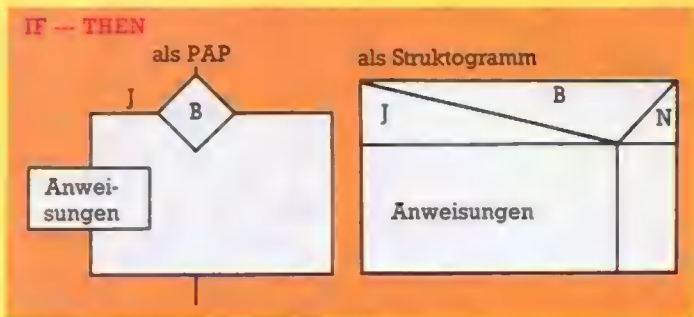


Bild 2. Die Anweisungen werden nur dann ausgeführt, wenn die Bedingung B erfüllt ist. ▲

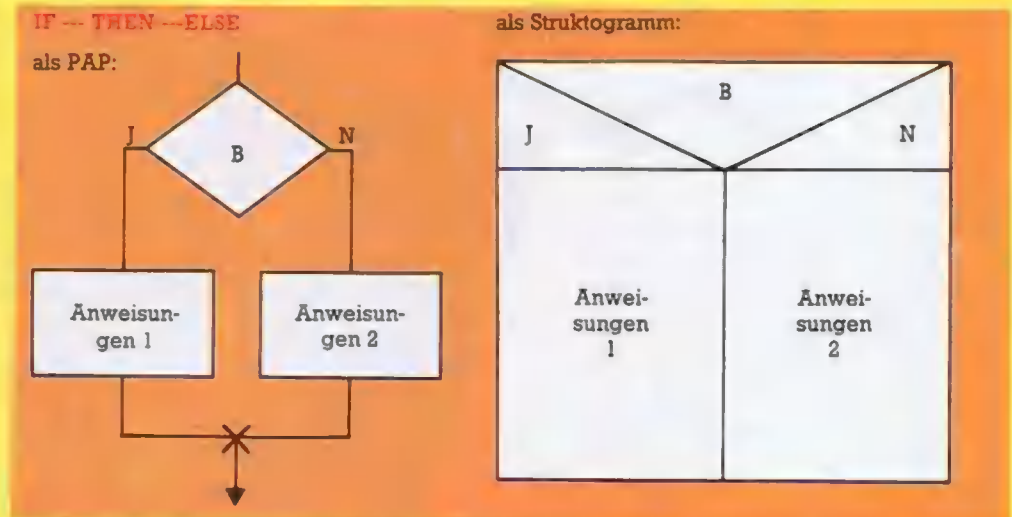
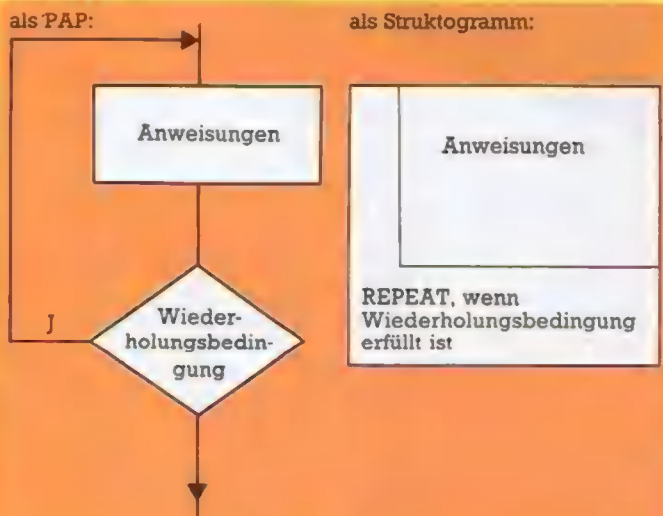
**REPEAT**

Bild 4. Die Anweisungen 1 werden nur dann ausgeführt, wenn die Variable den Wert 1 enthält, die Anweisungen 2 werden nur dann ausgeführt, wenn die Variable den Wert 2 enthält, die Anweisungen 3 werden nur dann ausgeführt, wenn die Variable den Wert 3 enthält, die Anweisungen n werden nur dann ausgeführt, wenn die Variable den Wert n enthält. Es können beliebig viele Blöcke von Anweisungen vorkommen. ►

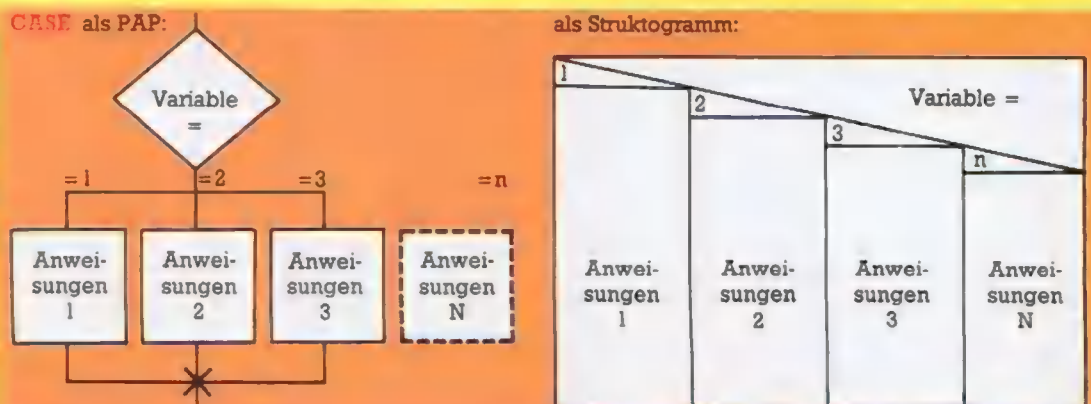


Bild 5. Die Anweisungen werden ausgeführt, solange (=while) die Wiederholungsbedingung erfüllt ist. Dabei wird jedesmal vor Beginn überprüft. Um eine Endlosschleife zu vermeiden, ist es erforderlich, daß die Wiederholungsbedingung innerhalb der Gruppe der Anweisungen verändert wird. ▲

EXBASIC


Schon längst gibt es Programme, die das Arbeiten mit dem Commodore 64 sämtliche Möglichkeiten, manche beschränken sich auf spezielle jedoch Schwerpunkte und besitzt auch Schwachstellen. Exbasic

Daß die auf Vielseitigkeit ausgelegte Hardwarekonzeption des Commodore 64 erst mit einer leistungsfähigeren Sprache beziehungsweise Spracherweiterung voll zum Tragen kommt, steht außer Zweifel. Das bei Commodore seit langem bestehende Mißverhältnis zwischen der hard- und softwaremäßigen Grundausstattung der Geräte war daher auch seit längerer Zeit für einige Software-Anbieter Anlaß, sinnvolle Erweiterungen des Betriebssystems und des Basic-Sprachumfangs zu entwickeln.

Exbasic Level II wurde ursprünglich für die älteren und größeren Commodore-Computer als fest zu installierender Hardware-Zusatz konzipiert. Die vorliegende C 64-Version kann diese Herkunft nicht leugnen, da sie nur bedingt an die neuen Möglichkeiten dieser Maschine angepaßt erscheint. Während zum Beispiel Simons Basic eindeutig neu für den C 64 entwickelt wurde und die grafischen und akustischen Fähigkeiten des C 64 voll unterstützt, hat man mit Exbasic Level II eher ein Produkt für kommerzielle Programme, also zum Beispiel solche aus dem Business-Bereich in der Hand.

Exbasic wird als Erweiterungsmodul (siehe Bild) zusammen mit zwei Handbüchern geliefert. Der Vorteil dieser, gegenüber einer reinen Diskettenversion, teureren Lösung spürt man deutlich, wenn man regelmäßig mit Exbasic arbeitet, da durch die langsame Übertragungsgeschwindigkeit vom Laufwerk zum C 64 für 8 KByte schon eine beträchtliche Ladezeit benötigt wird. Man steckt einfach das Modul in den Expansion-Port und erhält bereits nach dem Einschalten eine entsprechende Startmeldung. Es belegt

den Speicherbereich \$8000 bis \$9FFF, man hat also 8 KByte weniger Speicherplatz zur Verfügung als sonst.

Unbequemer ist da schon die Lektüre der beiden Handbücher: Das stärkere gelbe Buch beschreibt die Exbasic-Versionen für die älteren Commodore-Computer in einem zügig lesbaren, lockeren Stil, mit vielen kleinen Beispielen. Um aber mit der C 64-Version zurecht zu kommen, muß man gleichzeitig das blaue Ergänzungsbuch lesen. Darin findet man neben vielen Verweisen auf das Handbuch Hinweise zu Befehlen, die  in der C 64-Version nicht gibt, zu erweiterten Parametrisierungen von existierenden Befehlen und Beschreibungen von solchen Befehlen, die speziell für den C 64 dazugekommen sind. Man sollte sich also, um Frustration zu vermeiden, beim Studium von dem blauen Ergänzungsbuch leiten lassen. Die Befehlsübersicht in diesem Artikel soll den Überblick erleichtern.

Die einzelnen Gruppen, in die man die Befehle gliedern kann, sollen im folgenden besprochen werden. Dabei verdienen manche Befehle noch eine besondere Erwähnung

Hilfsfunktionen

Diese dienen dazu, das Erstellen und Testen von Basic-Programmen komfortabler zu gestalten. Es wurden dabei einige sinnvolle Eigenschaften realisiert, die sich nicht direkt durch Schlüsselwörter ausdrücken lassen. So kann man ein Programm komfortabel mit Hilfe der Cursortasten listen: Drückt man am unteren Bildrand die Cursor-down-Taste, so werden weitere Zeilen gelistet, während das Bild nach oben rollt. Für den oberen Bildrand und die Cursor-up-Taste gilt entsprechendes. Wird ein Programm fehlerhaft abgebrochen, so wird nicht nur eine Fehlermeldung abgesetzt, sondern auch die fehlerhafte Zeile mit dem Cursor an der fehlerhaften Stelle angezeigt. Diskettenbefehle werden abgekürzt (entsprechend den Möglichkeiten des DOS 5.1 auf der Demodiskette):



erleichtern. Manche unterstützen Bereiche. Jede Erweiterung setzt Level II von Interface Age ist dabei eine

LEVEL II

der interessantesten Spracherweiterungen für den Commodore 64.

/programmname	lädt ein Programm
!programmname	lädt ein Programm und startet ■
-programmname	speichert ein Programm auf Disk
@	fragt den Fehlerkanal des Laufwerks ab
@\$	zeigt das Inhaltsverzeichnis an, ohne das Programm im Speicher zu zerstören
@Befehl	sendet einen Befehl an das DOS

LOAD und SAVE bezeichnen Kassettenbefehle. Diese laufen in Exbasic mit fünffacher Geschwindigkeit ab.

FIND

Damit kann man alle Zeilen vollständig listen lassen, die einen angegebenen Suchstring an irgendeiner Stelle enthalten. So werden zum Beispiel mit FIND REM alle Kommentarzeilen gelistet. Mit FIND A\$ kann man sich alle Stellen, an denen auf die Variable A\$ zugegriffen wird, listen lassen. Zum Editieren von Programmen ist diese Möglichkeit von nicht zu unterschätzendem Wert.

RENUM

Exbasic unterstützt weniger als Simons Basic das strukturierte Programmieren. Sprünge und Aufrufe von Unterprogrammen sind immer mit der Angabe von Zeilennummern verbunden. Dem trägt aber der Befehl RENUM Rechnung. RENUM nummeriert nicht nur (wie bei Simons Basic) die Zeilennummern neu, sondern ändert auch die Argumente aller Befehle, die sich auf Zeilennummern beziehen, wie GOTO, GOSUB, THEN und so weiter, sinn gemäß ab, so daß, wie man es eigentlich auch erwartet, das unnummerierte Programm lauffähig bleibt.

TRACE

Dieser Befehl schaltet einen Modus ein, in dem jede aktuelle Befehlszeile eines laufenden Programms in der ersten Bildschirmzeile kurz angezeigt wird. Bei Zeilen mit mehreren Befehlen steht sogar

der Cursor auf dem jeweils aktuellen Befehl. So lassen sich zum Beispiel auch bedingte Anweisungen gut mitverfolgen. Verlangsamung des Vorgangs und auch Einzelschrittbetrieb sind mit der CTRL-Taste möglich. Durch Drücken der Commodore-Taste läuft das Programm wieder schneller. Da die Befehle TRACE und TRACE OFF auch programmierbar sind, kann man sich in einem Programm auch gezielt Teile, die man testen will, »tragen« lassen.

SPACE

Um Speicherplatz zu sparen, eliminiert Exbasic alle Blanks in Programmen außer in Kommentaren und in Texten, die durch Apostrophe begrenzt werden. Durch den Befehl SPACE (ohne Parameter) werden beim Listen (nicht im Programm selbst) wieder Blanks eingefügt, um die Übersichtlichkeit eines Programms zu wahren.

BASIC

Dieser Befehl deaktiviert Exbasic, was außer einem geringen Geschwindigkeitsgewinn bei Standard-Basic-Programmen eigentlich nichts bringt. Er wird allerdings notwendig, wenn man mit den Diskettenbefehlen LOAD"name".8 und LOAD"name".8,1 oder SAVE"name".8 arbeiten will, oder wenn das Programm diese Befehle enthält. Diese Befehle wirken in Exbasic un erfreulicherweise auf Kassette und nicht auf die Disk. Exbasic kann mit PRINT USR(0) wieder gestartet werden.

MERGE

Hiermit kann man ein Programm von Diskette oder Kassette zu einem Programm im Hauptspeicher dazuladen. Das nachzuladende Programm sollte Zeilennummern haben, die von denen des vorhandenen Programms verschieden sind, ansonsten bedarf das Mischprodukt noch einer manuellen Nachbehandlung, da doppelte Zeilennummern auftreten. Die richtige Form des Befehls für das Nachladen von Diskette ist übrigens:

MERGE* "programmname", 8

KEY

Durch die Möglichkeit, die f-Tasten mit beliebigen Texten zu belegen (einschließlich RETURN durch die Taste »-«) werden die guten Editierfähigkeiten von Exbasic abgerundet.

Kontrollstrukturen

Da Exbasic mehr das zeilennummern-orientierte Programmieren unterstützt findet man hier nur spärliche Möglichkeiten. Die von Pascal her bekannten Konstruktionen wie WHILE ... DO oder REPEAT ... UNTIL wird man hier vergeblich suchen. Das eigentlich schon zum Standard gehörende IF ... THEN ... ELSE gibt es aber auch in Exbasic.

DISPOSE

Der Nutzen dieses Befehls erscheint eher zweifelhaft. Er läßt zwar manchmal trickreiche und kurze Programme zu, untergräbt aber das Konzept von strukturierten Programmen. Der Interpreter verwaltet für Schleifen und Unterprogramm-sprünge einen Stack, auf dem Rücksprungadressen und Namen von Laufvariablen gehalten werden. Diesen Stack kann man mit DISPOSE manipulieren. So kann man mit DISPOSE NEXT eine Schleife abschließen, um sie dann mit GOTO irgendwo zu verlassen. DISPOSE RETURN schließt eine Unterprogrammebene ohne Rücksprung ab. Man muß dann das Unterprogramm mit einem GOTO verlassen. DISPOSE CLR löscht den gesamten Stack. Die Beispiele im Handbuch zu DISPOSE sind nicht sehr gelungen. Eine vernünftige Anwendung ergibt sich meiner Ansicht nach nur im Zusammenhang mit einer programmierten Fehlerbehandlung.

Fehlerbehandlung

Die Befehle zur Fehlerbehandlung sind sehr nützlich und sinnvoll.
ON ERROR GOTO

Wenn dieser Befehl einmal abgearbeitet wurde, bricht bei einem späteren Fehler das Programm nicht ab, sondern springt an die bei **ON ERROR GOTO** angegebene Zeilennummer. Dort kann der Benutzer seine eigene Fehlerbehandlungs-routine unterbringen. Dazu stehen ihm zur Information die Variablen **EC** (Error Code) und **EL** (Error Line) zur Verfügung. **EC** enthält die Nummer des Fehlers gemäß einer Liste im Handbuch, **EL** die Zeilennummer der Zeile, in der der Fehler aufgetreten ist. Die Fehlerbehandlungs-routine wird mit einer der drei **RESUME**-Varianten verlassen (siehe Tabelle).

Ein- und Ausgabebefehle

Diese machen neben den Hilfsfunktionen die eigentliche Stärke von **Exbasic** aus. Mit wenigen, aber leistungsfähigen Befehlen ist eine vielfältig formatierbare Ausgabe möglich. Man kann auch sehr einfach gegen Fehlbedienung gesicherte Eingabemasken realisieren.

INPUTFORM

Dieser Befehl gestattet es, neben der Ausgabe eines Textes, ein Eingabefeld in Position, Länge und Farbe zu definieren. Gleichzeitig wird mit diesem Befehl eine Eingabe über dieses Feld angefordert. Dabei ist es nicht möglich, über das festgelegte Feld hinauszuschreiben. Eingegeben werden können alle Zeichen außer den Cursor-Steuerzeichen. Mit **DEL** kann man einzelne Zeichen löschen, mit **»-«** den ganzen Text.

PRINT USING

Formatierte Ausgabe von Dezimalzahlen wird hiermit ermöglicht. Es wird ein Formatstring ausgegeben, der beliebig Text sowie Formatzahlen enthalten darf. Bei der Ausgabe werden dann die Formatzahlen durch aktuelle Zahlen aus der Variablenliste ersetzt. Eine Formatzahl setzt sich aus Formatzeichen **"#"**, **"*"**, **"."**, **"0"**, **"+"** und **"-"** zusammen. Dabei stehen **"#"** und **"*"** für Dezimalstellen. Beispielsweise führt **PRINT USING "DM # # #, # #", 2000/3** zu **DM 666,67**

Diese Befehle sind mit **Exbasic Level II** möglich

Konventionen: **X,X1,Y** und so weiter bezeichnen allgemeine numerische Ausdrücke, ansonsten werden Parameternamen klein geschrieben. **»ad«** steht für Adresse (0...65535), **»fz«** für Farbzahl (1...16). Runde Klammern müssen mit eingegeben werden, eckige und spitze Klammern werden nicht eingegeben. Parameter in eckigen Klammern müssen angeführt werden, während Parameter in spitzen Klammern optionell sind.

Hilfsfunktionen:

FIND (string)(,bereich)
AUTO (zeilennr.)(,schrittweite)
DEL (bereich)
RENUM (startnr.)(,schrittweite)
TRACE/TRACE OFF
DUMP
MATRIX
LETTER
LETTER OFF
MEM
HIMEM [ad]
SPACE/SPACE OFF
HELP
HELP*
BASIC
PRINT USR (0)
MERGE (,programmname)
KEY
KEY [nummer], [string]
KEY ON

Zeilen, die den String enthalten, listen
automatische Zeilennummernvorgabe
Bereich löschen
Zeilen unnummerieren
aktuellen Befehl anzeigen
nichtindizierte Variablen mit Wert anzeigen
indizierte Variablen mit Wert anzeigen
Umschalten in Klein/Großschrift
Umschalten in Großschrift/Blockgrafik
Speicherbelegung anzeigen
obere Speichergrenze für Basic setzen
beim Listen Blanks mit ausgeben
Exbasic-Schlüsselwörter auflisten
Standard-Basic-Schlüsselwörter auflisten
Rückkehr ins Standard-Basic
Exbasic Level II reaktivieren
weiteres Programm in bestehendes einkopieren
F-Tastenbelegung anzeigen
F-Tastenbelegung definieren
Standardbelegung herstellen

Kontrollstrukturen:

IF...THEN...ELSE
RESTORE [zeilennummer]
ON X RESTORE [zeilennummernliste]
DISPOSE
DISPOSE CLR
DISPOSE RETURN

DISPOSE NEXT [variablenname]

bedingte Verzweigung
DATA-Zeilen-Zeiger auf angegebene Zeile setzen

ON ERROR GOTO [zeilennummer]
RESUME

RESTORE mit berechneter Zeilennummer
Manipulationen des Stack
Abschluß aller Schleifen und Unterprogramme
Abschluß einer offenen Unterprogrammebene ohne Rücksprung
Abschluß der Schleife mit der angegebenen Laufvariablen sowie aller Schleifen innerhalb dieser, aber kein Sprung aus der Schleife

RESUME NEXT

Fehlerausgang definieren
Rückkehr nach Fehlerbehandlung zur Zeile, in der der Fehler aufgetreten ist
Rückkehr nach Fehlerbehandlung zu der Zeile, die auf die Zeile folgt, in der der Fehler aufgetreten ist
beliebiger Rücksprung nach Fehlerbehandlung

RESUME [zeilennummer]

Ein-/Ausgabebefehle:

INPUTLINE ("text";)
[stringvariable]
INPUTFORM ("text";)[stringvariable] (,maximallänge)(,fz)

Eingabe sämtlicher Zeichen

Eingabe beliebiger Zeichen außer Cursorsteuerzeichen in das Feld mit angegebener Länge

PRINT@ [position], "text"
PRINT USING ["format"], [varliste] (position;)
SPACE [spalte1,zeile1,spalte2,zeile2](,code)(,fz)

Ausgabe ■■■ beliebiger Bildschirmposition
formatiertes Drucken

STRING\$ (anzahl, string od. X)

Die ersten 4 Parameter bestimmen den linken oberen bzw. den rechten unteren Eckpunkt eines Rechtecks, das mit dem durch seinen Bildschirm-code gegebenen Zeichen gefüllt wird.
Vervielfachung des angegebenen Strings oder des durch den ASCII-Code **X** gegebenen Zeichens
Ausgabe des Bildschirminhalts (nur Blockmodus)

HARDCOPY

Stringbefehle:

INSTR (String 1, string 2 (position;))
EVAL (string)

EXEC (string)

sucht string2 in string1 ab position und liefert, falls vorhanden, die Position von string2 in string1, sonst 0
Auswertung eines durch string dargestellten numerischen Ausdrucks
Ausführung eines im string enthaltenen Basic-Befehls



Jet Set Willy KOMMT !

Das neueste Grafik-Abenteuer '...

Englands grösstes Ereignis seit den Beatles!

Willy, der legendäre Bergarbeiter aus MANIC MINER, hat es geschafft! Er ist reich, hat ein riesiges Haus, eine eigene Yacht und viele neue "Freunde". Nach einer durchzechten Nacht muss er erst einmal alle Gläser und Flaschen aufsammeln, bevor er in sein französisches Bett sinken kann.

WER HILFT IHM DABEI?

* * *

COMPUTER PLUS SOFT

verlost unter den ersten 10 Einsendern, die die Frage: "Wieviel Gläser und Flaschen muss Willy einsammeln?" richtig beantwortet haben, einen Drucker EPSON RX 80 F/T, 6 Champagnergläser mit der dazugehörigen Kiste Champagner (vom feinsten) und ein Autogramm von Matthew Smith, Englands Programmierer Nummer EINS.

Der Rechtsweg ist dabei wie immer ausgeschlossen!

Viel Glück wünschen die
COMPUTER PLUS SOFT Mitarbeiter

Exclusiv in Deutschland

COMPUTER PLUS SOFT

BAHNSTR. 22-26

4220 DINSLAKEN

☎ : 02134/7905

Händleranfragen erwünscht



BESTELL-COUPON

Jet Set Willy

KASSETTE: -COMMODORE 64 JE 45,-DM

-SPECTRUM 48K JE 39,-DM

DISKETTE: -COMMODORE 64 JE 49,-DM

		STCK.
		STCK.
		STCK.

ZZGL 3,-DM VERSANDKOSTEN

☐ PER NACHNAHME

☐ VERRECHNUNGSSCHECK
LIEGT BEI

DATUM UND UNTERSCHRIFT :

EXBASIC LEVEL II

Grafik- und Farbbefehle:

CURSOR [fz]	Ausgabefarbe setzen
GROUND [fz]	Hintergrundfarbe setzen
BORDER [fz]	Rahmenfarbe setzen
COKE [position,zeichencode,fz]	Durch Code bestimmtes Zeichen an Bildschirmposition setzen
DEEK [position,modus]	modus = S: Bestimmung des Zeichencodes des Zeichens an gegebener Position modus = C: Bestimmung des Farbcodes
HPlot X[,fz]	horizontalen Balken der Länge X plotten
VPlot X[,fz]	vertikalen Balken der Höhe X plotten
SET (spalte,zeile)	Punkt setzen (Blockgrafik 80x50)
RESET (spalte,zeile)	Punkt löschen
POINT (spalte,zeile)	Test, ob Punkt gesetzt

Tonerzeugung:

VOLUME X	Lautstärke des SID setzen (0-15)
ADSR [stimme, kurvenform,a,d,s,r],(pulsweite)	Klangparameter setzen
PLAY [stimme,tonhöhe]	Ton anschlagen

Numerische Funktionen:

MAX (X1,X2,...,Xn)	Maximum
MIN (X1,X2,...,Xn)	Minimum
FRAC (X)	Nachkommastellen einer Dezimalzahl
ROUND (X,Y)	Rundung von X auf Y Nachkommastellen
ODD (X)	Test des ganzzahligen Teils von X auf ungerade
RND (X)	Für X >= 2 ganze Zufallszahlen zwischen 1 und X, sonst wie Standard-RND
HEX\$ (X)	Umwandlung von X in Hexadezimalstring
DEC (Hexadezimalstring)	Umwandlung in Dezimalform

Verbindungen zur Maschinensprache:

CALL (parameterliste)	Aufruf eines Maschinenprogramms mit Parametern über speziellen Einsprungvektor
DEF CALL = [ad]	Definition des CALL-Vektors
DOKE [ad],X	Doppelbyte-POKE
DEEK (ad)	Doppelbyte-PEEK
VARPTR (variablenname)	Anfangsadresse, ab der der zugehörige Variablenwert im Speicher liegt

Sonstige Befehle:

SEC X	X Sekunden Verzögerung
PAUSE X	X/16 Sekunden Verzögerung
LOCK	Manuelle Umschaltung Schreibmaschinenmodus/Grafikmodus sperren
LOCK OFF	LOCK aufheben
SWAP [varname1],[varname2]	Werte zweier Variablen vertauschen

Auch wenn man keine EXBASIC-Level II-Befehle in seine Programme einfügen will, also nur das Standard-Commodore-Basic benutzt, ist EXBASIC-Level II eine sehr komfortable Editierhilfe

SPACE mit Parameterangabe

Im Gegensatz zu SPACE ohne Parameter (siehe oben) kann man mit SPACE plus Parameterangabe sehr schnell ganze Rechteckbereiche mit einem Zeichen beschreiben. Man kann damit zum Beispiel große farbige Rechtecke erzeugen oder Teilbereiche des Bildschirms gezielt löschen.

Stringbefehle

EVAL und EXEC erlauben es, Strings als numerische Ausdrücke beziehungsweise als Basic-Befehle

aufzufassen. Damit ist es möglich, in ein laufendes Programm eine Funktion einzugeben, um zum Beispiel deren Nullstellen zu ermitteln, ohne das Programm abubrechen. Denkbar wäre auch die Simulation eines leistungsfähigen Taschenrechners.

Grafik- und Farbbefehle

Es werden leider nur die Möglichkeiten unterstützt, die sich aus den Grafikzeichen der Tastatur ergeben. Dazu gehören Balken, deren Länge beziehungsweise Höhe sehr fein abgestuft werden kann, deren

Breite aber immer acht Rasterpunkte beträgt. Da diese Balkengrafik allerdings sehr schnell ist, könnte man sie beispielsweise zur quasi-analogen Anzeige von Meßwerten in Realzeit benutzen.

Weitere Befehle

Die Befehle zur Tonerzeugung ersparen eine lange Reihe von PEEKs und POKEs, auch muß man die Adressen der SID-Register nicht kennen. Man kommt jedoch nicht umhin, sich mit den Einzelheiten des Sound-Chips zu befassen. Eine Reihe weiterer Befehle erleichtert das Zusammenspiel zwischen Basic- und Maschinenspracheprogrammen erheblich. Viele Programmieraufgaben lassen sich nur in Maschinensprache vernünftig lösen. Diese Maschinensprache versteht man aber meistens mit Rahmenprogrammen in Basic, die dazu dienen, komfortabel mit dem Benutzer zu kommunizieren (in Basic liegen komfortable E/A-Routinen eben schon vor) und dazu, das Maschinenprogramm mit Parametern zu versorgen und es aufzurufen. Letzterem dienen die Befehle HEX\$, DEC, CALL, DOKE, DEEK und VARPTR. Man halte sich vor Augen, daß zum Beispiel zur Änderung eines 16-Bit-Wertes der Befehl:

DOKE AD,X genügt, während man in Standard-Basic die etwas umständliche Sequenz:
POKE AD,X-256*INT(X/256)
POKE AD+1,INT (X/256)
benötigt.

Zusammenfassende Beurteilung

Wer die vollen Grafik- und Klangmöglichkeiten des C 64 (einschließlich Sprites) mit von Basic ausnutzen will, der sollte Simons Basic oder eine andere spezielle graphische Befehlserweiterung wählen. Auch was strukturierte Programme angeht, fährt man mit Simons Basic besser.

Wer allerdings Basic-Programme (unter anderen auch schon vorhandene) sehr komfortabel editieren und testen will, wer maskierte Eingabe und formatierte Ausgabe benötigt, für Business-Programme, für komfortable und schnelle Rahmenprogramme zu Maschinenspracheprogrammen, für den ist Exbasic Level II sicher das geeignetere Werkzeug.

(Thomas Krätzig)

Welches Textprogramm

Die Verarbeitung von Texten ist ganz sicher eines der wesentlichen Einsatzfelder für Computer, und zwar sowohl im privaten wie auch im kommerziellen Bereich. Die Anforderungen, die dabei an die entsprechenden Programme gestellt werden, sind durchaus unterschiedlicher Natur.

zu welchem Preis?

Marktübersicht Textverarbeitung

Programme	Computer	Preis in DM	Anbieter	Adresse
Blitztext	C 64	199,-	Hofacker GmbH	8150 Holzkirchen, Tegernseerstr. 18
Vizawrite	C 64	298,-	Microton	CH-2542 Pieterlen, Postfach 40
Wordpro 3 Plus	C 64	285,-	Zoom Soft	6000 Frankfurt/M, Obermainstr. 30
Texted Direct	C 64	28,-	L. Himer	6308 Butzbach, Postfach 466
Edit + T	C 64 / VC20	ab 695,-	Rudolph EDV	2000 Hamburg 74, Münsterweg 6
SM Text 64	C 64	250,-	SM-Software	8000 München 83, Scherbastr. 33
Wordproc	C 64	139,-	Microcomp. Laden	1000 Berlin 12, Kantstr. 70
VIC Writer	C 64	98,-	RMC-Systems	4200 Oberhausen, Im Steinhaldchen 23
M&T Textverarb.	VC 20	129,-	Markt & Technik	8013 Haar, Hans-Pinsel-Str. 2
Text 64	C 64	189,-	Commodore GmbH	6000 Frankfurt, Postfach 710126
Homeword	C 64	239,-	Markt & Technik	8013 Haar, Hans-Pinsel-Str. 2
Textomat	C 64	99,-	Data Becker	4000 Düsseldorf, Merowingerstr. 30
Micro-G-Text	C 64 / VC20	69,-	Micro-G	7963 Eichstegen, Ortstr. 5
Text	C 64	198,-	Luther Elektronik	1000 Berlin 20, Heerstr. 340
Textlogig	C 64	280,-	Anubis GbR	7928 Gengen/Brenz, Schwagestr. 77
Hes Writer	C 64	129,-	Ariolasoft	8000 München 80, Steinhäuser Str. 3
64 Script	C 64	898,-	PDC Profess. Data	3000 Hannover 1, Heiligengeiststr. 15

Marktübersicht über die für den Commodore 64 / VC 20 angebotenen Textverarbeitungsprogramme. Für Ergänzungen ist die Redaktion jederzeit dankbar.

Der private Anwender, der seinen Computer als reines Hobby betrachtet, wird in der Regel nicht viel Geld für ausgefeilte Textprogramme ausgeben wollen und sich oft genug auch seine eigene Textverarbeitung schreiben. Anders sieht es zum Beispiel bei dem Geschäftsmann aus, der seine gesamte Korrespondenz mit Computerhilfe abwickeln will. Dieser Anwender wird naturgemäß viel Wert auf »gehobene« Funktionen zur Textverarbeitung legen und auch eher bereit sein, für wirklich gute Programme auch einiges an Geld zu zahlen.

Die vorliegende Marktübersicht enthält — ohne einen Anspruch auf Vollständigkeit zu erheben — die gängigsten der für den Commodore 64 (zum Teil auch für den VC 20) angebotenen Textverarbeitungsprogramme. Die Preise liegen dabei zwischen 28 und 898 Mark, eine Spanne, innerhalb der sowohl der reine Hobbyist wie auch der kommerzielle Anwender ein für ihn geeignetes Programm finden dürfte. Dabei muß ganz deutlich darauf hingewiesen werden, daß nicht für jede Anwendung das teuerste Programm auch unbedingt das beste sein muß. Zum Beispiel nützt einem die in einem teuren Programm eventuell in-

tegrierte Adreßverwaltung wenig, wenn man gar keine Adressen verwalten will. Man sollte sich wie immer beim Softwarekauf vorher überlegen, was man nachher braucht.

Was gibt es auf dem Markt?

Diese Marktübersicht kann Ihnen keine Beschreibungen der einzelnen Textprogramme liefern (wir müßten sonst ein ganzes Heft damit füllen), aber Sie erfahren hier, was Sie zu welchem Preis wo beziehen können — und vor allem, was für Programme überhaupt auf dem Markt sind. (ev)

SM-TEXT 64

Der Wunsch, mit dem Commodore 64 eine Verarbeitung von Texten durchzuführen, wird sicherlich in jedem Besitzer einen Commodore 64 spätestens dann wach, wenn er sein System um einen Drucker bereichert hat.

Da die Textausgabe mittels PRINT-Anweisungen dabei schnell als zu umständlich angesehen wird, steht man bald vor der Entscheidung, ein fertiges Textverarbeitungsprogramm zu kaufen. Der in der Reihe der »Golden Tools« Programme erschienene SM-Text, verspricht diese Aufgabe in komfortabler Weise übernehmen zu können.

Wer bereits beruflich oder während seines Studiums mit einem Textverarbeitungssystem gearbeitet hat, wird mehr verlangen, als allein die Fähigkeit, geschriebene Texte auf den Drucker zu übertragen:

1. Der Text sollte möglichst so auf dem Bildschirm erscheinen, wie er später gedruckt wird.
2. Das Einfügen, Verändern und Formatieren der Texte sollte unkompliziert und schnell sein.
3. Das Speichern und Laden der Texte muß möglich sein.
4. Die Fähigkeiten des angeschlossenen Druckers sollen ganz ausgenutzt werden.
5. Die Steuerung der einzelnen Funktionen sollte einfach, logisch und übersichtlich sein.
6. Letztlich sollte ein im deutschsprachigen Raum verkauftes Produkt auch die deutschen Umlaute darstellen und zum Ausdruck bringen können.

Einer Realisierung dieser Anforderungen stehen von der Hardware des Commodore 64 einige Probleme entgegen. Der 64er hat keine deutsche DIN-Tastatur als Standard, seine Zeichendarstellung ist auf den amerikanischen ASCII-Satz abgestimmt, und er kann nur 40 Zeichen pro Zeile darstellen.

Dieses Handicap wurde beim SM-Text auf beeindruckende Weise überwunden. Nach dem Laden des Programms erscheint ein Menü, über das sich entweder die Standard-Commodore- oder eine fast komplette deutsche DIN-Tastatur

auswählen läßt (Bild 1). Eine Anpassung an ASCII- oder Commodore-Drucker ist hier ebenfalls möglich. Je nachdem, welche der Funktionen angewählt wurde, speichert SM-Text die Tastatur/Drucker-Kombination auf der Systemdiskette ab. Dadurch wird erreicht, daß diese Einstellungen so lange automatisch mitgeladen werden, bis man eine andere Kombination wählt.

Verläßt man dieses Menü durch Drücken der Tasten »T« und RETURN erscheint nach zirka einer Minute das Arbeitsblatt-Hauptmenü (Bild 2). SM-Text ist aus über- und untergeordneten Menüs aufgebaut. Durch Drücken einer der Funktionstasten, deren Bedeutung am unteren Bildschirmrand erklärt wird, kommt man in das jeweils untergeordnete Menü, durch Drücken der »!«-Taste (Men) kommt man in das jeweils übergeordnete Menü.

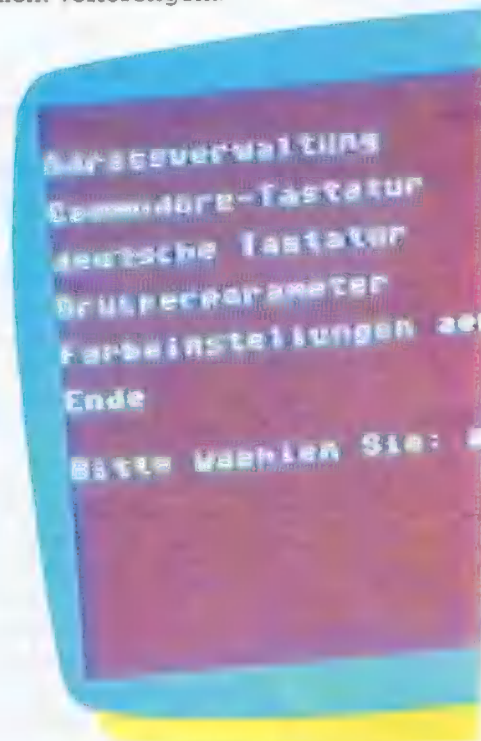
Da die Erklärungen der Funktionstasten am unteren Bildschirmrand relativ kurz gehalten werden mußten, besteht die Möglichkeit durch Drücken der CLR/HOME-Taste genauere Erläuterungen abzurufen und mit »Space« wieder zu löschen. So wie die Funktionstasten haben auch viele Steuertasten eine neue Aufgabe zugeordnet bekommen. Die »Pfeil-nach-links-Taste« wird zum Worttabulator (ein Wort vor, beziehungsweise zurück) und die RUN/STOP-Taste zum Tabulator. Mit CTRL in Verbindung mit dem Worttabulator, den Cursorsteuertasten und der SHIFT-Taste, lassen sich Wörter und Zeilen vor bzw. nach dem Cursor löschen.

Möchte man nun, ausgehend vom Hauptmenü, einen Text erstellen, so ist zuerst die fl-Taste (editieren) zu drücken. Der Cursor beginnt dann in der ersten Zeile auf einem Pfeil nach rechts zu blinken, die Anleitung sagt, daß man nun dem Text einen Namen geben soll. Dabei ist zu beachten, daß man den Cursor um eine Stelle nach rechts versetzt, damit der Pfeil erhalten bleibt. Macht man dies nicht, so erhält man die Fehlermeldung "Textname nicht in Ordnung". Hat man dies beachtet, drückt man RETURN und kann beginnen seinen Text zu schreiben.

die professionelle

Da ich die deutsche Tastatur gewählt hatte, erwartete ich mit Spannung die Darstellung der deutschen Umlaute: Und tatsächlich beim Drücken der <:>-Taste erscheint auf dem Bildschirm ein <ö>. Ermutigt von diesem Erfolg versuchte ich einmal, ohne auf die Tastatur zu schauen, alle mir von der Schreibmaschine her bekannten Zeichen zu erreichen. Das Resultat war erfreulich, alle Buchstaben waren da, wo sie auf einer deutschen Schreibmaschine auch sind. Wer also »blind« im Zehn-Finger-System schreiben möchte, und auf deutsche Umlaute nicht verzichten will, wird vom SM-Text freudig überrascht sein.

Aber wie sieht es mit der Darstellung des Textes auf dem Bildschirm aus? Auch hier wurde eine elegante Lösung gefunden: Der Text bewegt sich beim Schreiben in Sprüngen jeweils in der 36., 51. und 66. Spalte horizontal um einige Zeichen nach links. Die Sprünge wurden dabei aber so gewählt, daß ein Teil des eben geschriebenen Textes noch sichtbar bleibt, so daß der Anschluß nicht verlorengeht.



Eine der wichtigsten Anwendungen für Computer ist die Textverarbeitung. Daß eine sinnvolle und komfortable Textverarbeitung nicht nur mit zigtausend Mark teuren Geräten möglich ist, versucht SM-Text 64 zu zeigen.

Textverarbeitung

Möchte man aber, daß der gesamte Text auf dem Bildschirm lesbar ist, so kann man aus dem Editier-Modus über die f8-Taste die Zeilenbreite beliebig zwischen 0 und 120 Zeichen pro Zeile einstellen.

Auch das Verändern des geschriebenen Textes ist recht komfortabel gelöst: Zum einen bleiben die bekannten Cursorfunktionen ganz erhalten, zum anderen ist es weiterhin möglich, Texte zu überschreiben oder Wörter und Sätze einzufügen. Nach Drücken der SHIFT- und INST-Taste wird der bisher geschriebene Text automatisch so lange nach unten verschoben, bis eine der Cursor Tasten gedrückt wird.

Der nach dem Laden immer aktivierte Automatikmodus (reservs A auf der obersten Zeile) sorgt beim Schreiben dafür, daß Wörter, die nicht mehr ganz in eine Zeile passen würden, automatisch mit in die nächste Zeile übernommen werden. Man braucht also nur noch zu schreiben, ohne auf Worttrennung

gen beziehungsweise »Wagenrücklauf« wie bei der Schreibmaschine achten zu müssen. Eine Umformatierung des Textes ist über die Funktionen f7 (justieren) nach dem Schreiben immer noch möglich. Hilfreich sind dabei die sogenannten Cursorbewegungen (f1 im Editiermodus). Sie ermöglichen es, seitenweise vor oder zurück zu springen oder den Cursor an das Textende beziehungsweise den Textanfang zu setzen.

Auch das Löschen und Einfügen von Zeilen oder das Transportieren und Kopieren von einzelnen Bereichen (f5 im Editiermodus) ist leicht möglich. Wer dann noch Absätze justieren oder Zeilen zentrieren möchte, wird von SM-Text nicht enttäuscht.

Eine weitere, sehr nützliche Funktion ist das sonst nur von speziellen Textverarbeitungscomputern her bekannte Suchen im Arbeitsspeicher. Damit ist es möglich, nach bestimmten Wörtern im Text zu suchen und diese einmal oder immer durch ein anderes zu ersetzen. Besonders werden Verfasser von Rundschreiben diese Funktion zu schätzen wissen. Damit auch die für solche Rundschreiben notwendigen Adressen nicht extra eingetippt werden müssen, besteht ferner eine Schnittstelle zur SM-Adreva, einem anschließend erläuterten Adressenverwaltungsprogramm der gleichen Programm-Reihe.

Damit beim Laden, Löschen oder Abspeichern keine ungewollten Datenverluste entstehen, sind zahlreiche Sicherungen vorgesehen, zum Beispiel wird vor jedem Löschen des Arbeitsspeichers (f2 im Hauptmenü) gefragt, ob die Funktion auch wirklich ausgeführt werden soll.

Besonders gefallen hat mir, daß in der Funktion Laden (f3 im Hauptmenü) die Möglichkeit besteht, sich den Inhalt einer Diskette zeigen zu lassen und dann durch einfaches Drücken der bestimmten Nummer des Files dieses zu laden. Sollte der Arbeitsspeicher dabei nicht leer sein, wird dieser nicht überschrieben, sondern es wird eine Fehlermeldung angezeigt.

Leider sind mir beim Arbeiten mit SM-Text auch einige nachteilige Eigenschaften des Programms aufgefallen. So wird beispielsweise bei längeren Texten (mehr als 100 Zeilen) die Verarbeitungsgeschwindigkeit immer langsamer, bis sie bei voller Ausnutzung des Arbeitsspeichers fast zum Stillstand kommt. Es ist deshalb sinnvoll, immer nur eine Seite (66 Zeilen) zu schreiben, diese abzuspeichern, die Seitenzahl im Textnamen um eins zu erhöhen und die nächste Seite zu schreiben.

Da sich die, sonst sinnvolle, Eigenschaft des Programms, alle Texte so zu drucken wie sie auf dem Bildschirm geschrieben wurden, bei mehrseitigen Texten leider negativ auswirkt, ist es ratsam, jede Seite mit Leerzeilen aufzufüllen, damit beim Drucken auf Endlospapier nicht auf die Abreißkante geschrieben wird. Epson-Besitzer können sich natürlich auch hier mit einer Escape-Sequenz behelfen. Macht man dies, so kann man die Funktion Drucken (f7 im Hauptmenü) voll ausnutzen. Es wird einfach der Name des zu druckenden Textes in die oberste Zeile geschrieben und vom Hauptmenü aus die Taste f7 gedrückt. Der Text wird dann, direkt von der Diskette, Seite für Seite ausgedruckt.

Sehr zu loben ist das deutschsprachige Handbuch, das nur wenige Fragen offen läßt. Eine von diesen Fragen ist die Ansteuerung von anderen Schriftarten und Textformatierungsfunktionen, wie sie beispielsweise von Epson-Druckern verarbeitet werden können. Dort wird lediglich angedeutet, daß man durch Schreiben des !-Zeichens mit den nachgestellten Steuerzeichen diese Funktionen erreichen könnte.

Nach einigem Probieren ist es mir dennoch gelungen, alle Schriftarten der Epson-Drucker auf das Papier zu bringen. So wird beispielsweise durch ! ä die »Escape-Sequenz (CHR\$(27))« gesendet. Stellt man dahinter ein großes M, ist man im Schönschreibmodus. Oder durch ! ä-l wird der Unterstreichmodus eingeschaltet, durch ! ä-0 wieder ausgeschaltet. Den Besitzern des 1526-Druckers steht diese Funktion leider nicht zur Verfügung. Dafür



Bild 1. Hier können die Parameter eingestellt werden

SM-TEXT 64 • *die professionelle Textverarbeitung.*

können sie durch Drücken der »Commodore«-Taste zusammen mit einer Buchstabentaste den Grafik-Zeichensatz ausdrucken.

Insgesamt stellt SM-Text ein wirklich interessantes Textverarbeitungsprogramm dar, mit dem »fast« professionelles Arbeiten möglich ist. Da man die beschriebenen Nachteile durch einige kleine Tricks umgehen kann, ist ein komfortables Schreiben mit diesem Programm gewährleistet. Auch das Ausdrucken mehrerer Exemplare des gleichen Textes ist, obwohl dafür keine eigene Funktion vorhanden ist, durch entsprechend häufiges (maximal 10maliges) Drücken der f7-Taste, möglich.

Bei SM-Text ist es möglich, eigene Sicherheitskopien anzufertigen, was einen von der Angst befreit, das noch zirka 2500 Mark teure Programm durch einen Irrtum zu verlieren.

SM-Text-64 hebt sich von anderen Textverarbeitungsprogrammen vor allem dadurch ab, daß es für den Commodore 64 geschrieben wurde und dessen Nachteile auf sehr elegante Weise ausgleicht. Nach einiger Zeit vergißt man beim Arbeiten mit SM-Text, daß man nicht vor ei-

nem Textverarbeitungscomputer sitzt und das ist, so glaube ich, die beste Reverenz.

Die dazugehörige Adressverwaltung, SM-Adreva, kann natürlich auch ohne SM-Text verwendet werden, gewinnt aber wesentlich an Einsatzmöglichkeiten, wenn sie im Verbund mit SM-Text, beispielsweise zur Erstellung von Formbriefen, eingesetzt wird.

SM-Adreva, die »goldene« Adressverwaltung

Die Produktgestaltung ist recht ansprechend: Die Systemdiskette, ein gut gegliedertes, deutschsprachiges Handbuch, und, was keinesfalls eine Selbstverständlichkeit ist, ein Service-Scheck, mit dem man gegen Einsendung von 20 Mark eine zusätzliche Programmdiskette als Sicherheitskopie erhalten kann.

Erfreulicherweise wurden bei SM-Adreva alle vom SM-Text her bekannten Vorteile, wie die deutsche DIN-Tastatur, die Farbwahlmöglichkeit und die Anpassung der Druckerparameter (zum Beispiel an Epson-Drucker) mit übernommen.

Hat man nach dem Starten des

Programms die Einstellung der Drucker- und Tastaturparameter vorgenommen, so werden diese abgespeichert und sind fortan sofort nach dem Programmstart wieder eingestellt. Verläßt man dieses Menü mittels der Taste >A<, fordert das Programm den Benutzer auf, eine Datendiskette einzulegen. Da diese beim ersten Start natürlich nicht vorhanden ist, tut es zunächst auch eine leere, formatierte oder unformatierte Diskette. Die Formatierung wird dann vom Programm selbständig ausgeführt.

Nach der Formatierung der Diskette kann man sich ruhig erst mal eine Tasse Kaffee gönnen, denn nun beginnt das Programm auf der Diskette eine REL-Datei anzulegen, was alles in allem fünf Minuten dauert. Dieser Vorgang ist beendet, wenn die Bildschirmmaske erscheint. Auf dieser Datendiskette kann man nun bis zu 622 Datensätze (beziehungsweise Adressen) abspeichern. Sind mehr Datensätze zu verarbeiten, kann jederzeit eine weitere Datendiskette angelegt werden.

Die Bildschirmmaske des SM-Adreva ist vorgegeben und kann nicht mehr verändert werden, sie ist aber so angelegt, daß sie fast allen Ansprüchen gerecht werden kann (Bild 3). In der ersten Zeile befinden sich Kundennummer und Suchbegriff, in den darauffolgenden Zeilen Anrede, Name1, Name2, Zusatz, Straße, Ort und Briefanrede. Ferner besteht die Möglichkeit, Telefonnummer, Telexnummer und fünf Zeilen Text einzugeben (Bild 3).

Ganz am unteren Rand sind, wie in dieser Reihe üblich, wieder die Belegungen der Funktionstasten angezeigt. Mit Taste f1 werden Adressen gespeichert, mit f2 gelöscht, mit f3 gedruckt und mit f4 wird das Programm beendet. Die letzte Funktion ist nach Beendigung der Dateneingabe immer zu benutzen, da sonst die Dateien auf der Diskette nicht abgeschlossen werden und für immer verlorengehen können.

Grundsätzlich sind die Adressen über zwei Kriterien später wiederzufinden: Entweder nach »Kundennummer« oder nach »Suchbegriff«, wobei die Kundennummer nur Zahlen zwischen 1 und 622 annehmen

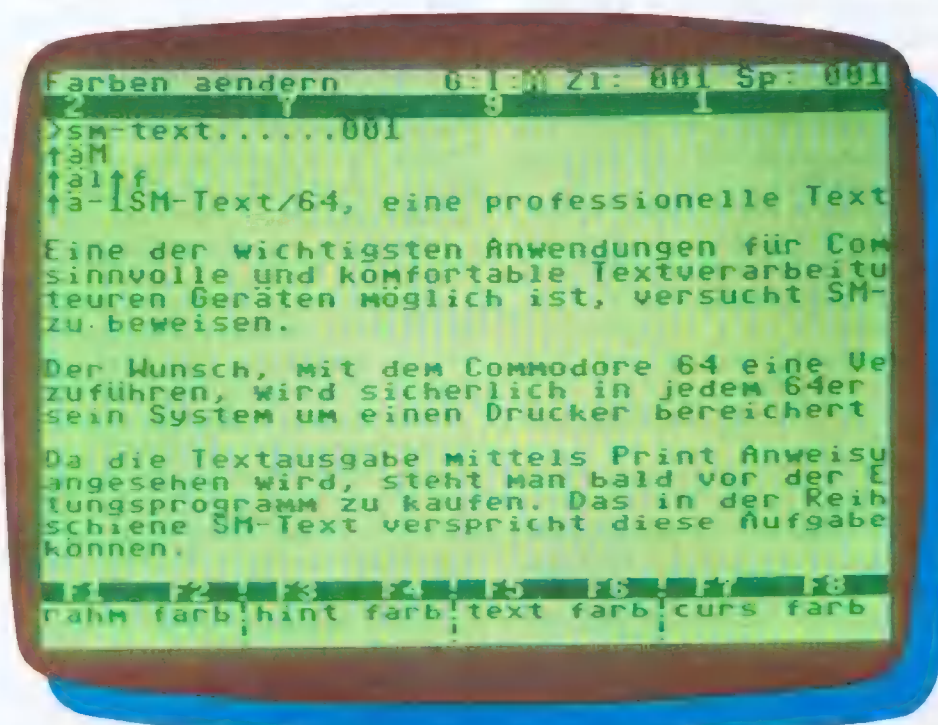


Bild 2. Das Arbeitsblatt-Hauptmenü mit Text

darf, der »Suchbegriff« dagegen Kleinbuchstaben und Ziffern mit bis zu zehn Zeichen.

Bei der Eingabe der Adressen ist es sinnvoll keine eigenen Kundennummern zu vergeben, sondern dieses dem Programm zu überlassen. Erst beim Suchbegriff kann man dann entweder einen Namen, eine Postleitzahl oder ähnliches eingeben. Hat man dies getan, so fragt das Programm: »Vor/Zurück/Ändern/Neueintrag?« Bei der ersten Eingabe ist natürlich immer <N> zu drücken. Jetzt ist es möglich, die Bildschirmmaske mit den gewünschten Daten auszufüllen. Die Cursorfunktionen bleiben dabei in vertikaler Richtung erhalten, horizontal (für Korrekturen) ist die DELETE-Taste zu verwenden. Sind alle Daten eingegeben, so kann der Datensatz durch Drücken der f1-Taste abgespeichert werden.

Möchte man aber einen bestehenden Datensatz verändern, so ist wahlweise der Suchbegriff oder die Kundennummer einzugeben und durch RETURN abzuschließen. Drückt man dann die Taste <A>, können die gewünschten Änderungen vorgenommen und abgespeichert werden (f1). Wenn aber der Suchbegriff oder die Kundennummer nicht richtig waren, so besteht die Möglichkeit, durch Drücken von <V> vorwärts und von <Z> zurückzublättern.

Besonders wichtig ist die Funktion des Druckens von Adressen. Dazu ist die betreffende Adresse zunächst zu suchen (wie bei Adresse ändern). Erscheint die gewünschte Adresse auf dem Bildschirm, kann sie durch f5 ausgedruckt werden. Oft ist es aber auch nötig, eine Adreßliste auszudrucken, dazu wird, ausgehend von einer leeren Bildschirmmaske, wieder die Taste f5 gedrückt. Nun besteht die Auswahl zwischen einer Liste und Aufklebern, bei einer Liste werden alle Daten der Datensätze ausgedruckt, bei Aufklebern nur Name und Anschrift.

Der Ausdruck erfolgt immer alphabetisch, wenn nach einem Suchbegriff geordnet gedruckt werden soll. Bei der Kundennummer beginnt der Ausdruck mit der ersten

Nummer und endet mit der letzten gespeicherten Kundennummer.

Die Erstellung von Formbriefen ist wohl eine der interessantesten Möglichkeiten der Adreßverwaltung. Hierzu ist aber, wie bereits gesagt, das Programm SM-Text notwendig. Ist dieses ebenfalls vorhanden, so kann nicht nur die Adresse in einen geschriebenen Text, sondern auch Daten, wie beispielsweise der Name, an beliebiger Stelle im Text eingefügt werden. Dazu muß beim Schreiben des Textes immer dann, wenn ein Name eingefügt werden soll, die Commodore-Taste und ein <A> geschrieben werden. Auf dem Bildschirm erscheint dann folgendes Zeichen: A, nun muß nur noch das betreffende Kürzel (zum Beispiel N1 für Vornamen) dahintergesetzt werden.

Zum Ausdruck ist das ganze dann dadurch zu bringen, daß durch Drücken der f4-Taste noch die Adressen hinzugeladen werden und, wenn die richtige Adresse vorliegt, durch anschließendes Drücken der <D>-Taste ausgedruckt wird.

Der größte Nachteil dieses sonst sehr ansprechenden Programms ist darin zu sehen, daß es nicht möglich

ist, sich beispielsweise alle Personen die, in einer Ortschaft leben oder deren Name mit A anfängt, ausgeben zu lassen, es sei denn, man hätte bei der Eingabe als Suchbegriff den Namen oder die Ortschaft gewählt. Da man damit aber ein für alle Mal festgelegt ist, bleibt oft nichts anderes übrig, als sich viele Adressen anzusehen, um die gewünschten Personen zu finden. Bei einer möglichen Datensatzmenge von 622 Einträgen sicherlich eine Menge Arbeit.

Dennoch stellt SM-Adreva (Preis: 175 Mark) ein insgesamt attraktives

Adressenverwaltungsprogramm dar, da eine echte deutsche Tastatur und Druckersteuerung verwirklicht wurde und nicht zuletzt auch deshalb, weil die Möglichkeit zur Erstellung von Formbriefen besteht. Da diese Funktionen aber leider nur zusammen mit dem zirka 250 Mark teuren SM-Text möglich ist, erreichen diese beiden Programme zusammen eine Preiskategorie, in die sich möglicherweise nicht mehr jeder Hobbyanwender vorwagt. Für eine wirkliche professionelle Adressenverwaltung reichen die beschriebenen Funktionen aber wahrscheinlich nicht aus. (Arnd Wängler)

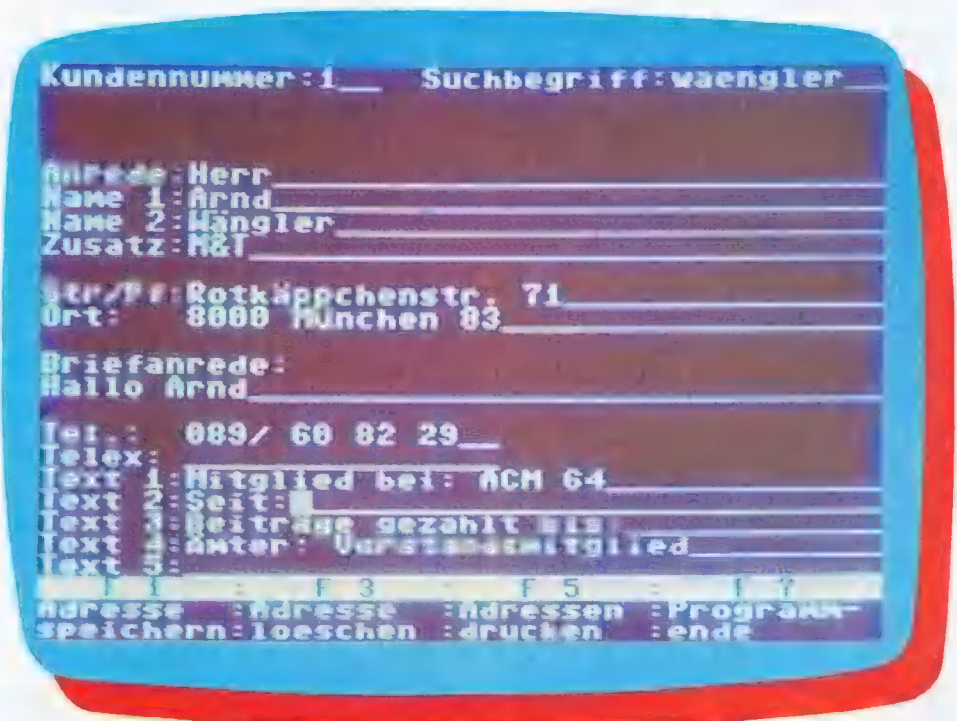
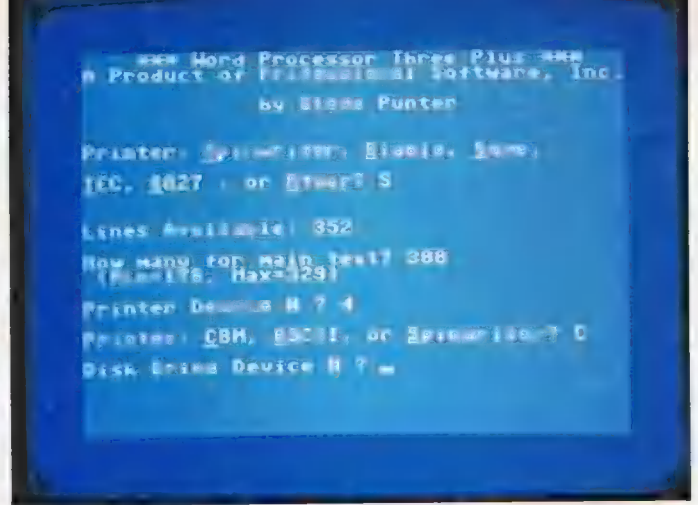


Bild 3. Viele Eingabemöglichkeiten sind bei Adreva in der Bildschirmmaske möglich

Von den Großen auf den Kleinen: Wordpro 3 Plus



Menü: Eingabe der Gerätenummer von Floppy und Drucker sowie Druckerauswahl

Nach dem Laden und Starten des Programms meldet sich der Computer mit:

*** Word Processor Three Plus ***
A Product of Professional Software,
Inc. by Steve Punter
Printer: Springwriter, Diablo, Qume,
TEC, 8027, or Other?

Druckerwahl

Wordpro 3 Plus ist mit den hier aufgeführten Druckern (und einigen anderen) kompatibel. Sie können nun Ihren jeweiligen Druckertyp wählen, indem Sie den entsprechenden Anfangsbuchstaben eingeben.

Anschließend erscheint eine andere Frage auf dem Bildschirm:
Lines Available: 352
How many for main text?
(Min = 176, Max = 329)

Dadurch erfahren Sie, daß im Arbeitsspeicher 352 Zeilen (je 40 Zeichen) für die Textaufnahme zur Verfügung stehen.

Mit der Frage »How many for main text?« begegnet man einer wichtigen Besonderheit von Wordpro 3 Plus. Der Arbeitsspeicher ist nämlich in zwei Bereiche aufgeteilt, einen für den Haupttext (main text) und einen für den Zusatztext (switch text). Im Hauptspeicher befindet sich der Text, den man gerade bearbeitet. Der Zusatzspeicher eignet sich für Notizen, öfter benötigte Textbausteine und vor allem für Adressen, die man im Serienbrief verwenden kann. Auch das Inhaltsverzeichnis der Diskette läßt sich hier unterbringen.

Das Programm verlangt also zwischen 176 und 329 Zeilen für den Hauptspeicher. Sollte man sich nicht für die maximale Angabe entscheiden, werden die restlichen Zeilen dem Zusatzspeicher zugeschrieben.

Dann erkundigt sich das Programm nach der Gerätenummer des Druckers und gibt weiter drei Druckertypen zur Auswahl. Auch nach der Gerätenummer der Flop-

py wird gefragt: Disk Drive Device #?

Achtung: Wordpro 3 Plus arbeitet nur korrekt, wenn ein serieller Drucker beziehungsweise Floppy angeschlossen ist. Wollen sie über einen IEEE-488-Bus eine Floppy oder einen Drucker betreiben, dann meldet sich das Programm lautlos ab!

Statuszeile

Haben Sie alle Eingaben korrekt erledigt, dann erscheint ein leerer Bildschirm mit Statuszeile und einem Cursor:

Wordpro 3 Plus :X:I:S:C:NC=1 L=1

Diese Statuszeile bleibt immer auf dem Bildschirm, auch wenn der Text nach oben beziehungsweise unten scrollt. Die Buchstaben X:I:S:C:N geben durch Aufleuchten an, in welcher der fünf Hauptbetriebsarten man sich befindet.

- X** = Man befindet sich im Nebenspeicher
- I** = Einfügen. Diese Betriebsart ermöglicht es Ihnen, Wörter, Zeilen und ganze Absätze in einen vorhandenen Text einzufügen
- S** = Umschalten auf Großschreibung
- C** = Control-Modus. Wird durch Drücken der CTRL-Taste ein- und ausgeschaltet. In dieser Betriebsart verwandeln sich die meisten Tasten in Befehlstasten
- N** = Zahleneingabe — In diesem Modus sorgt Wordpro 3 Plus automatisch dafür, daß Zahlenkolonnen dezimalgerecht untereinander geschrieben werden

C = 1 : L = 1. Diese Zähler zeigen an, in welcher Position sich der Cursor gerade befindet:

C = Angabe der Spalte

L = Angabe der Zeile

Bei Wordpro 3 Plus braucht man sich bei der Eingabe des Textes über dessen endgültige Gestalt noch keine Gedanken zu machen. Man hat vielmehr noch die Möglichkeit, nachträgliche Änderungen im Text vorzunehmen. Es können Buchstaben, Wörter, Sätze oder sogar ganze Absätze gelöscht, eingeschoben oder verschoben werden. Das

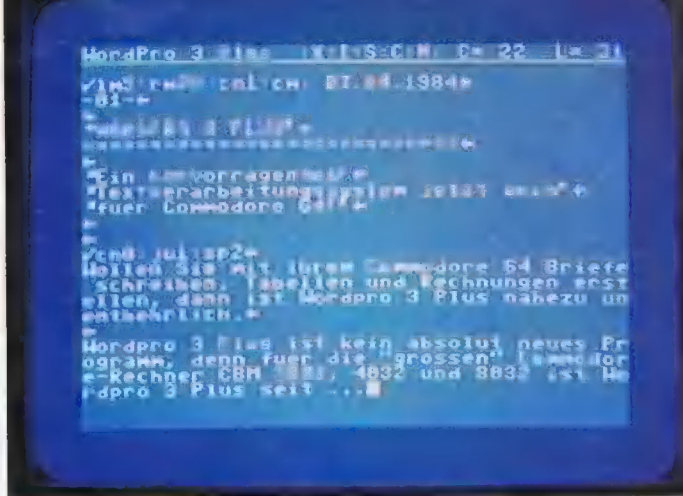
Bearbeiten eines Textes ist sehr einfach, da die Cursorsteuertasten wie vom Basic her gewohnt funktionieren. Die Return-Taste ist lediglich nach Absätzen oder Überschriften zur Textformatierung zu drücken.

Textformatierung

Durch Formatierungsbefehle gibt man anschließend dem Text seine äußerliche Gestalt. Die Formatierungsbefehle bestehen aus drei Zeichen und einer Zahl als Parameter. Das erste Zeichen CTRL und "/" gibt an, daß es sich hier um einen Formatierungsbefehl handelt, und daß der Rest dieser Zeile nicht gedruckt werden darf. Die folgenden zwei Zeichen stellen den Befehl dar.

Wordpro 3 Plus kennt insgesamt 23 Formatierungsbefehle: Hinter dem cm-Befehl kann ein Kommentar stehen; ähnlich dem Basic-Befehl »REM«. Folgende vier Befehle werden jeweils durch den nachfolgenden Parameter ein- (= 1) beziehungsweise ausgeschaltet (= 0). Der cn-Befehl zentriert alle Textzeilen zwischen dem linken und rechten Rand. Normalerweise ist der rechte Rand eines Textes »ausgefranst«.

Der Blocksatzbefehl ju ordnet den Text an beiden Rändern. Der ra-Befehl bewirkt nun, daß der rechte Rand bündig ausgedruckt wird und links ein »Flatterrand« entsteht. Durch den lf-Befehl können Sie einen zusätzlichen Zeilenvorschub bewirken. Den rechten und linken Rand Ihres Textes können Sie mit den beiden Befehlen rm und lm festlegen. Hier sind die Parameter (2-163) erlaubt. Geben Sie diese Befehle nicht ein, so setzt Wordpro 3 Plus den linken Rand auf Spalte 5 und den rechten Rand auf Spalte 75 (entspricht DIN-A4-Format). Der pp-Befehl meldet dem Drucker das Papierformat, das heißt wieviel Zeilen maximal zur Verfügung stehen. Der pg-Befehl hingegen gibt an, wieviel Zeilen auf einer Seite stehen sollen. Ähnlich wie bei der Schreibmaschi-



Wordpro 3 Plus im Eingabemodus

ne kann mit dem sp-Befehl der Zeilenabstand eingegeben und mit dem ma-Befehl der Rand gelöscht werden. Mit fp wird ein Seitenumbruch erzwungen.

Wollen Sie im Text eine entsprechende Anzahl Leerzeilen haben, so können Sie dies durch den ln-Befehl und der Angabe der Anzahl der Leerzeilen erreichen. Der vp-Befehl erzeugt automatisch den in der nachfolgenden Zahl festgelegten Zeilenvorschub vom Beginn jeder neuen Seite an. Für Schönschreibdrucker kann man die Schreibdichte pt und Zeilendichte fa jeweils in Zoll wählen. Müssen Sie während des Druckens zum Beispiel das Typenrad wechseln, so geben Sie den ps-Befehl und den entsprechenden Kommentar ein. Wenn der Computer auf dieses Zeichen stößt, hält er an, gibt den Kommentar in der Statuszeile aus und wartet auf das Drücken der »C«-Taste um fortzufahren.

Sie haben die Möglichkeit, bei längeren Dokumenten jeweils am Anfang einer jeden Seite eine Kopfzeile und am Ende eine Fußzeile zu drucken. Die Kopfzeile steht hinter dem Formatierungsbefehl hd und die Fußzeile hinter dem ft-Befehl. Der linke und rechte Rand der Kopfzeile kann separat mit hl beziehungsweise hr programmiert werden. Die Kopfzeile kann auch eine »mitzählende« Seitennummer enthalten. Wollen Sie allerdings, daß mit dem Zählen nicht bei 1 begonnen wird, sondern zum Beispiel bei 25, dann geben Sie einfach den Befehl p#25 ein.

Außer den Formatierungsbefehlen gibt es noch die Control-Befehle. Wurde die CTRL-Taste gedrückt, so befindet man sich im Controlmodus (C in der Statuszeile leuchtet auf), und die meisten Tasten werden zu Befehlstasten. Die »>-Taste schaltet den »Disk-Command« ein; das heißt man kann Disketten formatieren, Programme löschen oder umbenennen und so weiter.

Die »>-Taste fragt den Disk-Status ab. Durch »CTRL«-A können Sie häufig benötigte Textbausteine aus dem Nebenspeicher abrufen. Das O steht für Output und schaltet den Drucker ein. Mit »CTRL«-G kann der Cursor auf eine bestimmte Zeile gesetzt werden. Wenn Sie bei der Texteingabe »CTRL«-B drücken, so erscheint auf dem Schirm zunächst nur ein Grafikzeichen. Beim Ausdruck allerdings holt sich das Programm bei jedem B eine Textvariable aus dem Nebenspeicher. Dies eignet sich vor allem für Serienbriefe. So können zum Beispiel die Adressen mehrerer Personen im Nebenspeicher untergebracht werden. Das Programm holt sich dann bei jedem Ausdruck eine neue Adresse aus dem Nebenspeicher.

Controlmodus

Das Inhaltsverzeichnis der Diskette wird mit »CTRL«-0 in den Speicher gelesen. Die E-Taste schaltet auf Löschbetrieb. Bei nachfolgendem A wird alles gelöscht. Es gibt jedoch auch die Möglichkeit, selektiv nur bestimmte Worte, Sätze, Absätze oder Bildschirmzeilen zu löschen oder einfach alle Zeichen ab der Cursor-Position.

Wollen Sie einen bestimmten Absatz im Text verschieben, so können Sie ihn zuerst mit »CTRL«-T auf eine andere Stelle übertragen.

Sehr nützlich ist auch die »FIND-Funktion«, mit der man lokal (im Text) oder global (auf Diskette) bestimmte Zeichenketten suchen kann. Mit einer ähnlichen Funktion kann ebenfalls eine Zeichenkette gesucht und automatisch gegen eine andere ersetzt werden. Wurde bei der »FIND-Funktion« die gewünschte Zeichenkette gefunden, so bleibt der Cursor unmittelbar dahinter stehen. Mit »CTRL«-H setzt das Programm die Suche nach einem weiteren Auftreten der Zeichenkette fort. In jeder beliebigen Spalte des Bildschirms kann ein Tabulator gesetzt oder gelöscht werden.

Wordpro 3 Plus ist kein absolut neues Programm, denn für die »großen« Commodore-Computer CBM 3032, 4032 und 8032 ist es seit langem bekannt. Wer einmal damit Briefe und dergleichen geschrieben hat, wird sich nur sehr schwer von diesem komfortablen Programm trennen können. Es ist seit einiger Zeit auch für den Commodore 64 verfügbar.

Wordpro 3 Plus kann Zahlenkolonnen addieren und subtrahieren. Für diesen Zweck schaltet man mit »CTRL«-N auf Zahleneingabe um. Das N in der Statuszeile leuchtet auf. In dieser Betriebsart können nun Rechenoperationen durchgeführt werden.

Eine Diskette kann mit »CTRL«-RUN/STOP initialisiert werden. Um Text zu laden oder zu speichern, hat man »SHIFT«-CLR/HOME zu drücken. In der Statuszeile erscheinen dann folgende Befehle: »Recall, Memorize, or Insert«. Der Recall-Befehl lädt ein Textfile von der Diskette; der Memorize-Befehl speichert ein Textfile auf Diskette. Mit Insert schließlich kann ein Textfile von der Diskette in den Text eingefügt werden.

Viele dieser Control-Befehle sind noch weiter in Unterfunktionen aufgegliedert. Es erscheint hier jedoch nicht sinnvoll, alle diese Unterfunktionen aufzuführen.

Eine Änderung der Bildschirmfarben ist von Wordpro 3 Plus ebenfalls vorgesehen. Hierzu dienen die Funktionstasten:

F7 + F5 ändert die Rahmenfarbe
F7 + F3 ändert die Bildschirmfarbe
F7 + F1 ändert die Zeichenfarbe

Wordpro 3 Plus ist für 285 Mark erhältlich. Leider ist das Handbuch nicht im Preis inbegriffen. Für dieses sind noch extra 22 Mark zu bezahlen.

Fazit

Ich finde, daß man mit diesem Programm in kürzester Zeit optisch hervorragende Texte erstellen kann. Eine Vielzahl von Funktionen ermöglicht nach einer gewissen Eingewöhnungszeit eine sehr schnelle Texteingabe, auch in Verbindung mit speziellen Strukturen wie Tabellen oder ähnlichem. Die komfortablen Editiermöglichkeiten tragen einiges zur sehr guten Benutzerfreundlichkeit bei. Insgesamt also wirklich ein Textprogramm, dessen Anschaffung sich lohnt.

(Christian Quirin Spitzner)

Bliztext

schnell

Bliztext möchte ein einfaches aber wirkungsvolles Werkzeug für die Erledigung der persönlichen Schreibarbeiten sein. Kann es diese Anforderung erfüllen?

Die Voraussetzung dafür, daß es dem Commodore 64 zusammen mit einem Textverarbeitungsprogramm gelingt, die gute alte Schreibmaschine in die Ecke zu verbannen, ist klar: Ein solches System muß mindestens die gleiche, möglichst noch eine höhere Leistung und besseren Bedienungskomfort bieten.

Bliztext verspricht dabei, dem Bediener in allen Situationen hilfreich zur Seite zu stehen, wenn er erst einmal mit der Bedienung dieses Wortprozessors vertraut ist. Daß dieser Vorgang des Kennenlernens eine umfangreiche Angelegenheit ist, merkt der interessierte Benutzer spätestens dann, wenn er das Handbuch zum ersten Mal durchblättert. Dort findet er auf fast jeder Seite eine Fülle von Steuerzeichen, die es zu beherrschen heißt. Von dieser Informationsfülle sollte man sich aber nicht verwirren lassen, denn im täglichen Gebrauch werden selten alle Befehle benötigt. Außerdem können noch einige Tricks die Arbeit er-

leichtern. Es ist deshalb zunächst sinnvoll, dem, glücklicherweise stufenweise aufbauenden Handbuch schrittweise zu folgen.

Erhältlich ist Bliztext sowohl als Disketten- als auch als Kassettenversion, wobei man bei der letzteren natürlich die Funktionen des Kommandokanals # 15 der Diskettenstation nicht nutzen kann; ganz abgesehen von der geringen Arbeitsgeschwindigkeit der Datasette. Aber auch die Besonderheiten der Diskettenstation, wie beispielsweise das Einlesen von Files während der Textformatierung, drängen zur Verwendung des VC 1541-Laufwerkes.

Das erste Bild, mit dem sich Bliztext nach dem Laden beim Benutzer meldet, ist die Texteingabemaske. Sie besteht am oberen Rand aus einer sogenannten Statuszeile, die Auskunft gibt über die Anzahl der geschriebenen Zeichen pro Zeile, den Gesamttext, den verbleibenden Speicherplatz und über den Zustand des Kopierregisters, dessen Funktion später noch erläutert wer-

den soll. Eigentlich ist der Wortprozessor nun bereit, einen Text aufzunehmen und es kann auch schon geschrieben werden. Dafür bietet der Speicher Platz für 27000 Zeichen, das entspricht etwa 7 bis 8 Schreibmaschinenseiten.

Das Schreiben

Die Eingabe von Texten gestaltet sich einfach, denn man braucht nur darauf los zu schreiben. Der Text schiebt sich dabei langsam in horizontaler Richtung von rechts nach links weiter, bis, ab der 75. Spalte, ein Warnton dazu auffordert, mittels Return in eine neue Zeile zu springen. So fährt man fort, bis der gesamte Text eingegeben ist. Hätten wir nun noch unsere alte Schreibmaschine, wäre der Text unveränderlich fertig und selbst das Einfügen eines einzelnen Wortes sehr schwierig. Nicht so bei Bliztext; es bietet noch einige Möglichkeiten, mit den eingegebenen Buchstaben umzugehen. Natürlich ist es machbar, den Text jetzt schon auszudrucken. Damit beraubt man sich aber der Besonderheiten, die einen Text erst ansehnlich und richtig machen.

Haben sie sich verschrieben? Keine Angst, durch Verwendung der Cursortasten auf die falsche Stelle fahren und verbessern. Auch wenn die Korrektur mehr Platz in Anspruch nimmt als der ursprüngliche Text, das Einfügen der zusätzlichen Zeichen ist kein Problem, denn der Text verschiebt sich automatisch nach rechts beziehungsweise nach unten. Auch ist es möglich, den ganzen Text nach einzelnen Worten zu durchsuchen und diese, falls gewünscht, durch andere zu ersetzen.

Das Erstellen von Serienbriefen wird somit wesentlich erleichtert. Natürlich kann man auch ganze Textblöcke markieren, löschen oder verschieben. Dies geschieht mit Hilfe des oben genannten Kopierregisters, in das bis zu 4000 Zeichen eingespeichert und bei Bedarf, an jeder Stelle des Textes wieder aufgerufen werden können.

<E> Setze die Tabulatorbreite
 Ein Zeichen zurück
<C> Change (Suchen und Ändern)
<D> Lösche letztes Zeichen
<E> Lösche das nächste Wort
entsprechend dem mitgegebenen
String
<F> Ein Zeichen vorwärts
<G> Füge Kopierregister ein
<H> Füge Hexbyte ein
<I> Füge String ein
<J> Springt zum Kommandozeilenanfang
<K> Lösche Textbuffer!!!
<L> Gebe formatierten Text auf ein Gerät aus
<M> Gebe formatierten Text auf einen Commodoredrucker
aus
<P> Senden Kommando(s) an Diskette
<R> Lese File ein
<S> Suche einen String
<T> Gehe in Terminalmode
<W> Schreibe File

Bild 2. Die Kontrollkommandos

wie der Blitz?

Eine Besonderheit des Blitztext Wortprozessors ist der sogenannte »Formatter«. Es handelt sich dabei um eine Funktion, mit der dem Text die Form gegeben werden kann, in der er später auf dem Drucker erscheinen soll. In diesem Formatiermodus ist es möglich, Texte zu zentrieren, Zeilen- und Seitenlängen zu definieren, einen rechten Randausgleich (Blocksatz) durchzuführen und Absätze hervorzuheben. Ferner

Umfangreiche Formatierungsbefehle

kann eine Seitennumerierung eingestellt werden und Steuerkommandos an den Drucker und die Floppy übermittelt werden. Somit ist es möglich, die vorhandenen Funktionen des jeweiligen Druckers anzusteuern, wie beispielsweise Fettdruck, Unterstreichen und Großschrift bei EPSON-Druckern. Besonders hervorzuheben ist die Fähigkeit, durch Formatierbefehle festzulegen, daß Files, die auf Diskette beziehungsweise Kassette abgespeichert sind, in bestehende Texte eingefügt werden.

Damit sind die Sonderfunktionen des Blitztext aber bei weitem noch nicht erschöpft. Ein sicherlich nicht unwichtiges Unterprogramm ist die Verwendung der Kommandozeile, die sich am unteren Bildschirmrand befindet. Eine Abbildung der verfügbaren Befehle ist in Bild 1 dargestellt. Diese reichen vom Setzen der Tabulatorbreite bis zum Terminalmode.

Der Terminalmode

Wer sich unter diesem Begriff augenblicklich noch nichts vorstellen kann, wird beim Studium des Handbuchs über diese Funktion überrascht sein. Im Terminalmode ist es dem C 64 möglich, mit der Außenwelt in Verbindung zu treten. Sei es mit anderen Computern direkt, oder via Modem und Telefon mit weiter entfernten Computern. Das

Übermitteln und Empfangen von Texten und Daten von einem Ort zum anderen wird dadurch unterstützt. Damit bei der Herstellung der Verbindung kein unliebsamer Datenverlust entsteht, ist die notwendige Hardware und deren genauer Anschluß glücklicherweise im Handbuch beschrieben. Etwas handwerkliches Geschick und technisches Verständnis muß aber dennoch vorausgesetzt werden.

Die zum Test vorliegende Version des Blitztextes war zusätzlich zu den oben genannten Funktionen um zwei weitere Unterprogramme gegenüber der früheren Version erweitert worden. Diese sind zum einen die Eingabemöglichkeit von Überschriften und Fußnoten, und zum anderen eine Rechenfunktion, mit der im Text Additionen und Subtraktionen ausgeführt werden können.

Leider ist bei so viel Licht auch immer etwas Schatten. Dieser wird insbesondere dadurch deutlich, daß sich die Programmierung der vielen Sonderfunktionen mitunter relativ umständlich gestaltet. Da große Teile des Programmes durch CTRL-Sequenzen (Bild 2) gesteuert werden und oft die Eingabe mehrerer dieser Sequenzen notwendig sind, kommt man gelegentlich um ein wahres Tastaturstakkato nicht herum. Umgebar ist dieser Arbeitsaufwand, zumindest teilweise, indem Files mit den bevorzugten Formatierungsbefehlen angelegt werden. Ferner ist es empfehlenswert, sich eine eigene Referenzliste aller Befehle anzufertigen, da diese im Handbuch leider nicht abgedruckt ist. Nicht zuletzt wegen der vielen

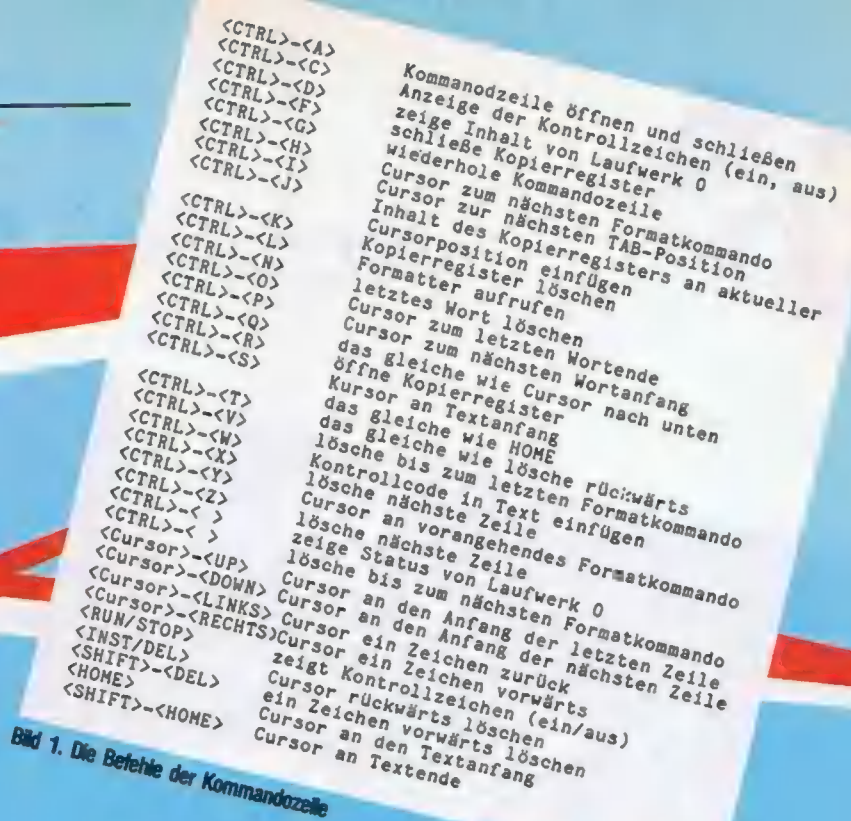


Bild 1. Die Befehle der Kommandozeile

Auswahlmöglichkeiten, die Blitztext bietet, ist eine sichere Bedienung des Programms wahrscheinlich erst nach einiger Übung zu erreichen, dann aber stehen einem Möglichkeiten zur Verfügung, mit der die anfangs erwähnte Schreibmaschine sicherlich nicht aufwarten kann. Letztlich bleibt nur zu wünschen, daß auch bei diesem Programm der deutsche Zeichensatz auf dem Bildschirm noch realisiert wird, auf dem Drucker können diese Zeichen durch Eingeben der entsprechenden Zeichen (zum Beispiel eckige Klammer links für das ö) umgangen werden.

Fazit

Blitztext ist ein leistungsfähiges Textverarbeitungsprogramm, dessen Bedienung zwar etwas umständlich und gewöhnungsbedürftig ist, dessen Leistungskriterien diesen Nachteil aber sicher ausgleichen. Besonders hervorzuheben ist die Möglichkeit der Datenübertragung zu anderen Computern. Diese ermöglicht zum Beispiel eine »orts-unabhängige Texteingabe« in der Bahn oder im Flugzeug auf sogenannten Handheld Computern wie dem TANDY-Modell 100 und der späteren Übertragung des Textes auf den C 64. Gemessen am Preis von 199 Mark kann dem Blitztext ein sehr gutes Preis-Leistungsverhältnis bestätigt werden, das sich dadurch noch verbessern läßt, daß man seine alte Schreibmaschine verkauft.

(Arnd Wängler)



- ein

mit **Pfiff**

Action-Spiel



Zwei Spielsituationen von AE auf dem VC 20

Gesamtüberblick:

Idee:	gut
Grafik:	sehr gut
Sound:	gut
Schnelligkeit:	sehr gut
andauernde Spielmotivation:	sehr gut
Gesamturteil:	sehr gut

Sie kommen aus den unendlichen Weiten des Weltalls. Retten Sie die Erde vor außer Kontrolle geratenen Robotern!

Die Roboter (AEs) wurden ursprünglich ins Universum geschickt, um in einer multinationalen Aktion die Welt zu retten. Die AEs sind jedoch außer Kontrolle geraten und greifen nun die Erde an.

Sie haben jetzt die Aufgabe, die Angriffe der AEs abzuwehren und diese in den Weltraum zurückzudrängen. Die Roboter greifen immer als Formation an. Hat man drei dieser Formationen bis auf den letzten AE ausgeschaltet, so kommt man ins nächste Schlacht-Bild. Sie haben nur wenig Zeit eine Formation völlig zu zerstören, denn die AEs verlassen nach kurzer Zeit das Schlachtfeld wieder. Aber kaum haben diese das Feld geräumt, ist auch schon eine neue vollständige Formation zur Stelle, um den Angriff fortzuführen.

Zu steuern ist das Spiel nur mit Joystick. Mit dem Steuerknüppel werden die Raketen-Batterien nach links beziehungsweise rechts bewegt. Das Abfeuern der Raketen erfolgt mit der Auslösetaste. Doch hier liegt eine Raffinesse des Spiels. Die Raketen detonieren erst nach dem Loslassen der Taste. So muß der Spieler die Raketen genau in der Flugbahn der AEs zum Detonieren bringen. Dies erfordert außerordentliche Umsicht und schnellstes Reaktionsvermögen, da die AEs zu allem Überfluß noch eigene Geschosse abfeuern, denen man ausweichen muß.

Das Spiel besteht aus vier Schlacht-Bildern, welche die grafischen Möglichkeiten des VC 20 voll ausschöpfen. Auch die vier Tongeneratoren des Volkscomputers bleiben

nicht unausgenutzt. Hat man einmal alle vier Schlachtfelder überwunden, so beginnt das Spiel von vorne, wird aber schwieriger, da die AEs noch trickreicher angreifen.

Das Spiel A.E. von Broderbund Software ist zum Preis von zirka 109 Mark als ROM-Modul für den VC 20 erhältlich.

Fazit:

Wer sich ein Schießspiel zulegen will, bei dem man nicht blind und monoton, sondern gezielt schießen muß, der wird bei diesem Spiel bestimmt nicht so schnell die Lust verlieren. Ein gutes Schießspiel zeichnet sich eben durch eine dauerhafte Spielmotivation aus.

(Christian Quirin Spitzner)



Dieses Spiel weist zehn verschiedene Szenen mit unterschiedlichen Schwierigkeiten auf und ist für 125 Mark im Handel erhältlich.

Laut Anleitung sind Sie nach dem Laden des Programms von Diskette »Time-Master-Time« und müssen mit Hilfe einer Zeitmaschine Dinosaurier-Eier und Jung-Dinosaurier aus der Steinzeit in die Zukunft retten.

Der Anblick des ersten Screens läßt einen glauben, daß man es mit einem weiteren Kletter-Hüpf-Renn-Spielchen zu tun hat. Doch das täuscht.

Dieses Spiel ist weitaus komplexer aufgebaut. Sie selbst erscheinen in Level 0 in einem Zeittor und müssen jedesmal, nachdem Sie so viel Eier, wie Sie tragen können — insgesamt drei Stück — gesammelt haben, in das Tor zurückkehren. Sie schicken die Eier in die Zukunft, fliegen selber mit und erscheinen wiederum irgendwo auf der Szene. Auch haben Sie die Möglichkeit, ab-

sonderlich mutierte Blumen zu fressen, »Power Flowers« genannt, die Ihnen die Kraft geben, mehr als drei Eier zu tragen.

An der Ausführung dieser Handlungen werden Sie von wie wild herumtrippelnden und schleichenden Spinnen, Schlangen und Tausendfüßlern gehindert. Diesen, sonst so possierlichen Tierchen, können Sie durch Herabrollen von heimtückischerweise an der Decke angebrachten Felsbrocken ein Ende machen, was einen dicken Pluspunkt entspricht. Bei Berührung derselben wird Ihnen jedoch Lebensenergie abgezapft und Sie müssen sich schleunigst in Ihr Zeittor zurückziehen, um diese wieder auf 100 Prozent zu bringen. Ist dies nicht der Fall, verwandeln Sie sich in eine Spinne, worauf Ihr zweites Leben beginnt.

Weiterhin versucht immer wieder der Fuß, der um Ihre kleinen Nachfolger besorgten Dinosauriermutter, der fast ein Drittel des Bildschirms ausfüllt, Sie zu zerquetschen. Daraufhin bricht Ihr drittes Leben an.

Hervorragend gemacht ist die Punktezahl, die nicht — wie bei anderen Spielen — in Tausendern oder Millionen zählt, sondern in realistischen Einschritten. Da sind 100 Punkte schon eine echte Leistung! Außerdem gibt es noch die schöne Möglichkeit, sich in höhere Levels zu teleportieren. Es werden aber die, in diesem Level noch nicht gesammelten Eier, als Minuspunkte abgezogen, wodurch es schon einmal passieren kann, daß man mit 250 Minuspunkten im Level 10 ankommt.

Insgesamt ist es eines der schönsten und einfallsreichsten Spiele der letzten Zeit, bei dem nur der Sound nicht ganz dem Niveau entspricht.

(Andreas von Lepel)



Sequenz

Eine Sequenz liegt vor, wenn mehrere Anweisungen hintereinander ausgeführt werden sollen (Bild 1).

In die jeweiligen Kästchen trägt man jeweils einen Befehl ein oder auch die Bezeichnung kompletter Programmteile.

Auswahl

Man unterscheidet drei Formen: **IF.THEN / IF.THEN..ELSE / CASE**

Man benutzt diese Formen, um Befehle oder Befehlsgruppen oder auch komplette Programmteile nur dann ausführen zu lassen, wenn eine Bedingung erfüllt ist (Bild 2, 3 und 4).

Wiederholung

Man benutzt zwei Formen der Wiederholung:

DO WHILE / REPEAT

Man benutzt diese Formen, um Befehle oder Befehlsgruppen oder auch komplette Programmteile mehrmals hintereinander ausführen zu lassen. Eine Programmschleife kann nur mit diesen Formen realisiert werden (Bild 5 und 6).

Facit

Beim Struktogramm entfällt völlig die Flußlinie, sie wird ersetzt durch die gemeinsame waagerechte Begrenzungslinie. Struktogramme kann man immer in Programmablaufpläne übertragen, umgekehrt ist es oft nur unter erheblichem Aufwand möglich. Das bedeutet, daß man durchaus auch mit Flußdiagrammen strukturiert programmieren kann; der Zwang dazu entfällt jedoch. Bei der Planung seiner Programme mit Struktogrammen wird man es schwer haben, »Spaghetti-Kode« zu produzieren. Und das ist allemal ein Vorteil. Und machen Sie bitte nicht den Fehler wie so viele (ich schließe mich dabei nicht aus) und erstellen Sie Ihr Flußdiagramm oder Struktogramm erst dann, nachdem Sie kodiert haben und Sie durch Ihr eigenes Programm nicht mehr durchsteigen. (gk)



WER HILFT DER JUGEND BEI DER ENTWICKLUNG?



COMMODORE COMPUTER.

Der Commodore-Heimcomputer berechnet, listet, und er bringt sogar technische Zeichnungen auf den Fernseher daheim. Ein tolles Ding: ein echter Computer mit unbegrenzten Möglichkeiten.

Er bringt aber auch riesigen Spaß für die Freizeit. Macht Musik, führt die Bundesligatabelle und spielt die spannendsten Videospiele. Ein faszinierendes Ding: ein echter Computer, den man spielend beherrscht. Der Commodore-Heimcomputer – der beliebteste Entwicklungshelfer der Jugend.

Beim Commodore-Vertragshandel, in führenden Warenhäusern, guten Rundfunk- und Fernsehfachgeschäften und großen Versandhäusern.

Mehr Informationen gibt's von: Commodore Büromaschinen GmbH, Abt. MK, Lyoner Straße 38, 6000 Frankfurt 71. Die Anschrift des Commodore-Fachhändlers in Ihrer Nähe erfahren Sie telefonisch von den Commodore-Verkaufsbüros:

Düsseldorf 02 11/31 20 47/48, Frankfurt 06 11/6 63 81 99, Hamburg 0 40/21 13 86, München 0 89/46 30 09, Stuttgart 07 11/24 73 29, Basel 0 61/23 78 00, Wien 02 22/67 56 00.

COMMODORE AUF VIDEO:

„Einblick für Leute mit Weitblick“

Über 1 Stunde spannende Informationen, wie ein Computer funktioniert und was man alles damit machen kann. Video Cassette (180er Scotch Band) per Nachnahme oder per Scheck anfordern bei:

Commodore GmbH - Video -
Postfach 260, 6082 Walldorf
(Achtung: bitte Video-System angeben!) Einmalige Schutzgebühr incl. Versand
zuzügl. nur 25,-
Nachn.-Geb. DM



Commodore

Eine gute Idee nach der anderen.

Quicktext ist ein vielfältig einsetzbares Textverarbeitungsprogramm für den Commodore 64 in Verbindung mit der Floppy 1541 und einem Matrixdrucker (zum Beispiel Epson MX-80/82, FX 80 mit Data Becker Interface). Das Programm benötigt zirka 12 KByte Speicherplatz; es bleiben also etwas mehr als 25 KByte zur Texteingabe übrig. Der einzige Nachteil ist eine relativ langsame Textausgabe auf dem Drucker, die sich aber durch Compilieren des Programms beschleunigen läßt.

- Zeilenlänge frei definieren
- Vorgegebenes Textformat übernehmen
- Neue Texteingabe
- Zeilen ändern, einfügen, löschen und kopieren
- Buchstaben oder Wörter im Text suchen und gegen neue Wörter oder Buchstaben austauschen
- Beliebiges Ausdrucken von Texten (Zeilen oder Blöcke)
- Variabler Zeilenabstand (1, 1.5, 2)
- Textformatierung (rechter Randausgleich)
- Seitennumerierung
- Briefkopf und Adresse nach DIN 676
- Unterstreichen, Fettdruck und Breitschrift in allen möglichen Kombinationen
- Wahlweiser Ausdruck auf Endlospapier oder Einzelblatt
- Druck der Sonderzeichen ä, ö, ü, Ä, Ü, Ö, ß
- Druck von hoch- und tiefgestellten Textteilen
- Umfangreiche Diskettenbefehle (siehe Bild 1).

Nach dem Laden und Starten meldet sich Quicktext mit dem Hauptmenü (Bild 2). Von dort aus werden die einzelnen Unterprogramme sowie das Diskettenmenü ausgewählt. Kehrt das Programm nicht von selbst ins Hauptmenü zurück, so ist dies jederzeit durch Betätigen der »Pfeil-Links-Taste« möglich (auch während der Texteingabe). Erläuterungen zu den einzelnen Unterprogrammen:

1. Text eingeben

Zunächst fragt das Programm, ob man neuen Text eingeben oder an einem bereits im Speicher befindli-

chen Text weiterschreiben will. Danach wird die Zeilenlänge bestimmt. Nach Eingabe der Zeilenlänge schaltet der Computer auf den Textmodus um, was am Farbwechsel des Bildschirms und dem neuen Cursor zu erkennen ist. Jetzt kann mit der Texteingabe begonnen werden. Der Wagenrücklauf wird mit der »Return«-Taste betätigt oder von selbst ausgelöst, wenn die maximale Zeilenlänge erreicht ist.

Die Cursorsteuertasten funktionieren im Texteingabemodus nicht. Der Textcursor kann mit Hilfe der IN- und DEL-Taste rückwärts bewegt werden. Größere Fehler können aber immer noch mit Hilfe des Editierprogrammes korrigiert werden.

Zur Eingabe von Sonderzeichen und Sonderfunktionen wird die Commodore-Taste in Verbindung mit der jeweiligen Buchstabentaste gedrückt (Bild 3).

Nach Beendigung der Texteingabe erfolgt die Rückkehr in das Hauptmenü durch Drücken der »Pfeil-Links-Taste«.

2. Text ändern

Zunächst fragt das Programm nach dem Textbereich, der gezeigt werden soll. Dieser wird jeweils zusammen mit zwei Befehlszeilen am unteren Bildschirmrand angezeigt. Die einzelnen Editierfunktionen sind in Bild 4 erläutert.

3. Text formatieren

Der Text wird auf Basis der eingegebenen Zeilenlänge formatiert (rechter Randausgleich).

4. Text ausdrucken

Das Programm fragt den Benutzer nun nach seinen Wünschen hinsichtlich Briefkopf, Zeilenabstand, Seitennumerierung, Länge des Ausdrucks, Linker Randabstand, Endlospapier oder Einzelblatteinzug.

In einem weiteren Durchgang werden nochmals alle gewünschten Funktionen angezeigt. Änderungen der Parameter sind vor der Ausgabe möglich. Dies ist wichtig, falls zum Beispiel vergessen wurde, den Briefkopf einzugeben. Nach Beendigung der Druckausgabe fragt das Programm, ob noch ein weiterer Druck gewünscht wird. Wenn ja, so

Quicktext

Daß eine gute Textverarbeitung nicht unbedingt mehrere hundert Mark kosten muß, zeigt das selbstgeschriebene Programm Quicktext. Auf dem Commodore 64 mit der entsprechenden Hardware bietet es den Komfort, den man manchmal bei den »Professionellen« vermißt.

geht es zurück zu den Einzelfragen. Wenn nein, so kehrt man ins Hauptmenü zurück.

5. Diskettenverwaltung

Vom Diskettenmenü (Bild 1) aus werden die einzelnen Unterprogramme ausgewählt, die in Bild 5 erläutert werden.

6. Briefkopf eingeben

Es gibt zwei Möglichkeiten, einen Briefkopf einzugeben: mit oder ohne Adreßfeld für einen Fensterbriefumschlag nach DIN 676. Das Programm fragt nach der gewünschten Form der Eingabe des eigentlichen Briefkopfes. Soll die betreffende Zeile des Briefkopfes oder der Adresse im Adressenfenster leer bleiben, so ist einfach die »Return«-Taste einzugeben.

7. End

Ausstieg aus dem Programm. Diese Funktion ist mit Vorsicht zu genießen, denn ein eventuell vorhandener Text bleibt zwar im

Speicher zunächst erhalten; wird Quicktext jedoch neu gestartet, werden alle Variablen (auch der Text in Z\$) unwiederbringlich gelöscht.

Für das einwandfreie Funktionieren des Programms sind natürlich noch einige Dinge zu berücksichtigen:

1. Die maximale Zeilenlänge beträgt 80 Zeichen. Bei der Eingabe von Sonderfunktionen erscheinen auf dem Bildschirm Grafikzeichen, die zwar nicht abgedruckt, jedoch im Text mit abgespeichert werden. Bei der Eingabe von Sonderfunktionen erhöht das Programm die Zeilenlänge jeweils entsprechend, aber nur bis maximal 80 Zeichen/Zeile.

2. Wenn der Zeilenabstand über mehrere Zeilen hinweg verändert werden soll, so ist das Sonderzeichen an den Beginn jeder Zeile zu stellen. Dies ist zum Beispiel bei längeren Zitaten erforderlich,

C= + a > ä	C= + z > Text
C= + o > ö	C= + x > Text
C= + u > ü	C= + c > Text
C= + s > ß	C= + v > Text
C= + linke eckige Klammer > {	C= + b > Text
C= + f > ö	C= + n > Text
C= + rechte eckige Klammer > }	C= + m > Text
	C= + d > Text
	C= + f > Text

Bild 3. Sonderzeichen und Sonderfunktionen

die sich über mehrere Zeilen hinziehen. Das Unterprogramm in den Zeilen 8500 bis 8560 verändert dann die Variable t (Zeilen pro Seite).

3. Das Programm akzeptiert nur die Eingabe von maximal 500 Zeilen. Sind 90 Prozent des Speichers vollgeschrieben, kehrt das Programm ebenfalls ins Hauptmenü zurück. Der Text kann anschließend noch geändert und abgespeichert werden.

4. Die RUN/STOP-Taste ist durch einen POKE-Befehl in Zeile 10 ausgeschaltet, um zu vermeiden, daß das Programm durch versehentliches Betätigen dieser Taste aussteigt.

5. Die Verwendung von Anführungszeichen ist nicht möglich. Sie werden durch das Hochkomma (') gesetzt.

Editiermöglichkeiten

-A- Das Programm fragt nach der zu ändernden Zeile. Diese wird auf den Bildschirm

gebracht und kann nun beliebig geändert werden. Das Fragezeichen und die Anführungsstriche dürfen nicht geändert werden, da sonst Fehler auftreten. Mit der »Return-Taste« kehrt man ins Hauptmenü zurück.

-L- Die betreffende Zeile wird gelöscht.

-E- Das Programm fragt, vor welcher Zeile eingefügt werden soll und schaltet dann in den Texteingabemodus um. Es können nur 60 Zeilen auf einmal eingefügt werden. Rückkehr mit der »Pfeil-Links-Taste«.

-T- Hier kann ein Buchstabe oder ein Wort im ganzen Text ausgetauscht werden. Das Programm fragt bei jeder einzelnen Möglichkeit, ob ausgetauscht werden kann. Ist dies nicht möglich, so wird »Tausch nicht möglich« eingeblendet. Wird bei der Bereichseingabe '0' eingegeben, so wird der gesamte Text nach Änderungsmöglichkeiten durchsucht.

-B- Beim Blocktausch können ganze Textblöcke mit beliebiger Länge versetzt werden. Auch eine Kopie ist möglich, wenn der alte Block einfach nicht gelöscht wird. '+' Mit dieser Taste wird der Text nach vorne geblättert. '-' Mit dieser Taste wird der Text zurückgeblättert.

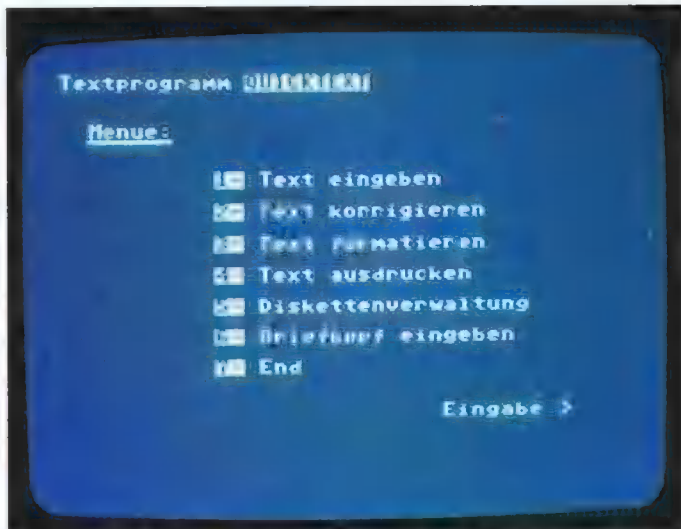
Mit der »Pfeil-Links-Taste« schließlich kehrt man ins Hauptmenü zurück.

Diskettenfunktionen

1. Text speichern

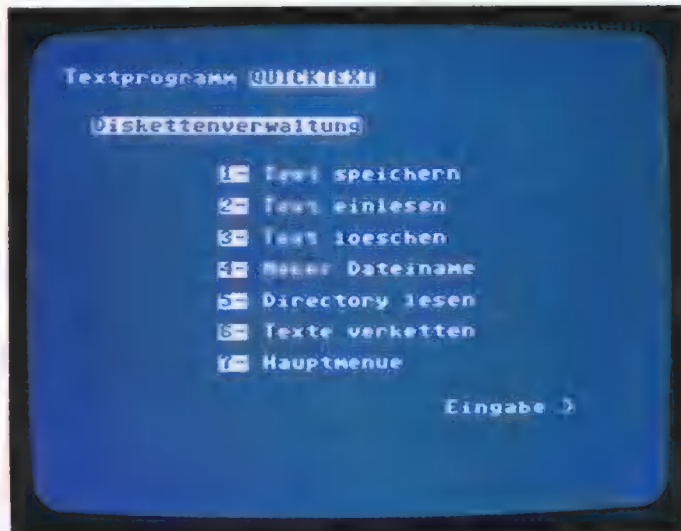
Das Programm initialisiert die Diskette, liest den Fehlerkanal aus und fragt dann den

Benutzer nach dem Dateinamen. Anschließend wird der Fehlerkanal wieder ausgelesen. Wenn alles in Ordnung ist, wird der Text gespeichert und das Programm kehrt ins Hauptmenü zurück. Ist ein Text mit gleichem Dateinamen vorhanden, wird gemeldet: »File exists«, und das Programm fragt, ob die alte Datei überschrieben werden soll. Wenn ja, wird die alte Datei überschrieben und durch die neue ersetzt. Wenn nein, muß man sich einen neuen Dateinamen überlegen.



▼ Bild 1. Diskettenmenü von Quicktext

▲ Bild 2. Hauptmenü von Quicktext



Zeilennummer	
10- 20	Bildschirmfarben, Dimensionierung, Initialisierung
30- 100	Sonderzeichendefinition; Sprung zum Hauptmenü
200- 405	Steuerung und Abfragen des Eingabecursors
2000- 2620	Texteingabe
3000- 3170	Edit-Bildschirm und Auswahl der Edit-Funktionen
3200- 3220	Änderung einer Textzeile
3300- 3310	Löschen einer Textzeile
3400- 3510	Einfügen einer Textzeile
3600- 3690	Wort- oder Buchstabenaustausch
3800- 3875	Blocktausch
4000- 4620	Textformatierung (rechter Randausgleich)
5000- 5220	Eingabe der Druckparameter
5230- 5395	Nochmaliges Anzeigen der Druckparameter
5400- 5560	Druck des Briefkopfes
5600- 5880	Druckausgabe
6000- 6070	Diskettenmenü
6200- 6250	Speichern eines Textes
6400- 6460	Einlesen eines Textes
6600- 6620	Löschen eines Textes
6800- 6820	Neuer Dateiname
7000- 7100	Directory lesen
7200- 7600	Texte verketteten
8000- 8100	Bildschirmfarben
8500- 8560	Änderung der Variable t (Zeilen pro Seite)
9000- 10922	Sonderzeichen und Sonderfunktionen für Druckausgabe
9100- 9110	Auslesen des Fehlerkanals der Diskette
9200- 9250	Datei auf Diskette überschreiben
10000-10080	Hauptmenü
11000-11110	Eingabe des Briefkopfes

Bild 5. Programmbeschreibung anhand der Zeilennummern

u	Zeilenlänge
z	Gesamtzahl der Zeilen
z\$(x)	Zeilennummer
t\$	normale Get-Abfrage nach Zeichen
fr	freier Speicherplatz
l	Anzahl der geschriebenen Zeichen
t	Zahl der Zeilen pro Seite
a	erste Zeile, die ausgedruckt werden soll
b	letzte Zeile, die ausgedruckt werden soll
p	Druckstartposition
za	Zeilenabstand
ma\$	Variable für Sonderzeichen
ul	Seitenzähler
ma	Zeilenzähler
ip\$	zu druckendes Zeichen
cb\$	Dateiname

Bild 6. Zusammenstellung der wichtigsten Variablen

August 6/June 1924


```

5180 PRINT "UNTERBLATTEINZUG" ; ENDLOSAPFIR
5181 GETF2$=IFF2$="" THEN5182
5184 IFF2$="B" ORF2$="E" THEN5190
5186 PRINT "Q":GOTO5180
5190 PRINT "WELCHE ZEILEN SOLLEN AUSGEDR
UCKT WERDEN?" ; GOTO5180
5200 INPUT "WIE KLEINERE ZEILEN:":A:IFA=0T
HENA=1:B=7:J=1:GOTO5230
5210 IFA<0ORA>ZATHENPRINT"00":GOTO5200
5220 INPUT "WIE HOCHERE ZEILEN:":B:IFB=AO
RB>ZATHENPRINT"00":GOTO5220
5230 IFW$="J" THENPRINT"GRIFKOPF WIRD A
BGEDRUCKT"
5240 IFW$="N" THENPRINT"GRIFKOPF WIRD N
ICHT ABGEDRUCKT"
5260 PRINT "WELCHEN ABSTAND ":IFA=36 THEN
PRINT "1"
5270 IFA=54 THENPRINT"1 1 5"
5280 IFA=72 THENPRINT"1 2"
5290 PRINT "LINIEN ABSTAND " ; P
5300 IFJ=1 THENPRINT"ABSDRUCK ALLER ZEIL
EN":GOTO5330
5310 PRINT "ABSDRUCK VON ZEILE" A " BIS Z
EILE" B
5320 IFAB$="J" THENPRINT"ZEITENNUMERIERU
NG"
5330 IFAB$="N" THENPRINT"ZEINE ZEITENNUM
ERIERUNG"
5340 IFF2$="B" THENPRINT "UNTERBLATTEINZ
UG"
5350 IFF2$="E" THENPRINT "UNTERLOSAPFIR"
5360 PRINT "WELCHE EINGABEN RICHTIG? (J/N
)"
5370 GETF1$=IFF1$="" THEN5370
5380 IFF1$="J" THENPRINT"ABSDRUCK ERFOLG
T. LITTE WARTEN WIE":GOTO5400
5390 IFF1$="N" THENP=0:A=0:B=0:F$="":GOTO
5000
5395 GOTO5370
5400 OPEN1,4,1:OPEN2,4,7:PRINT#1,CHR$(27
)"D"CHR$(P):CHR$(0);:IFW$="N" THEN5600
5410 PRINT#1,CHR$(27)"E" SPC(6) W1$ SPC(64-
(LEN(W1$))) W4$
5420 PRINT#1,SPC(6) W2$ SPC(64-(LEN(W2$)))
W5$
5430 PRINT#1,SPC(6) W3$ SPC(64-(LEN(W3$)))
W6$:PRINT#1,SPC(6) W8$
5440 IFA<1<>" THEN5470
5450 PRINT#1,CHR$(13) CHR$(27)"F":PRINT#1
,SPC(73) W7$
5460 FORX=1 TO3:PRINT#1,CHR$(13):NEXT:GOT
O5600
5470 FORX=1 TO2:PRINT#1,CHR$(13):NEXT:PRI
NT#1,CHR$(27)"F"
5480 PRINT#1,CHR$(15) SPC(15) W1$ * "W4$"
* "W5$CHR$(18):PRINT#1,SPC(8);
5490 FORI=1 TO40:PRINT#2,"":NEXT
5500 PRINT#1,CHR$(13)
5510 FORP=1 TO5: IFA$(P)=" " THEN5540
5520 IFA$=54 THENPRINT#1,CHR$(13) SPC(8) A$(P
):GOTO5560
5530 PRINT#1,SPC(8) A$(P)
5540 NEXTP
5560 FORY=1 TO3:PRINT#1,CHR$(13):NEXT:PRI
NT#1,SPC(74) W7$:PRINT#1,CHR$(13)
5600 MA=0:U1=1:IFA$=36 THENRB=4
5603 IFA=54 THENRB=1
5606 IFA=72 THENRB=2
5610 IFF2$="E" ANDAB$="N" THENRB=RB+1
5620 IFW$="N" THEN5650
5630 IFW$="J" ANDA$(1)=" " THEN5640
5635 IFA=36 THENT=4
5636 IFA=54 THENT=2
5637 IFA=72 THENT=20
5638 GOSUB5800:A=X+1:GOTO5700
5640 IFA=36 THENT=5
5641 IFA=54 THENT=34
5642 IFA=72 THENT=26
5643 GOSUB5800:A=X+1:GOTO5700
5650 IFA=36 THENT=64
5651 IFA=54 THENT=43
5652 IFA=72 THENT=32
5653 GOSUB5800:A=X+1:GOTO5700
5700 MA=0:U1=U1+1
5710 IFF2$="B" THENPRINT"LEITE"U1" EINLEG
T"(J/N):;INPUTF3$:IFF3<"J" THEN5710
5720 IFF2$="E" THENFORR=1 TO8:PRINT#1,CH
R$(13):NEXTR
5730 IFAB$="J" THENPRINT#1,SPC(35)"-U1"-
"CHR$(13)
5740 IFB=36 THENT=62
5741 IFB=54 THENT=41
5742 IFB=72 THENT=30
5750 GOSUB5800:A=X+1:GOTO5700
5800 FORX=1 TO8:PRINT#1,CHR$(9)*FOR77=IT
OLEN(Z$(X)):IP$=MID$(Z$(X),ZZ,1)
5805 IFIP$="" ORIP$=" " THEN5830
5808 IP=ASC(MID$(Z$(X),ZZ,1))
5810 IFIP=176 ORIP=185 ORIP=184 ORIP=91 ORIP
=168 ORIP=92 ORIP=174 THEN9000
5820 IFIP=173 ORIP=187 ORIP=188 ORIP=190 ORIP
=191 ORIP=170 ORIP=167 ORIP=154 THEN9020

```

[illegible]

```

9000 POKE$289,9:P0#E532B1,0:RETURN
9100 POKE$289,12:P0#E532B1,2:RETURN
9520 IF ZB=36ANDZA=54THENT=T-.3
9510 IFZB=36ANDZA=72THENT=T-.6
9520 IFZD=54ANDZA=72THENT=T+.3
9530 IFZB=54ANDZA=72THENT=T-.3
9540 IFZB=72ANDZA=54THENT=T+.3
9550 IFZB=72ANDZA=54THENT=T+.3
9560 RETURN
9000 IFIP=176THENMAI=AET
9002 IFIP=185THENMAI=DEI
9004 IFIP=184THENMAI=UEI
9006 IFIP=91THENMAI=GAI
9008 IFIP=168THENMAI=GII
9010 IFIP=93THENMAI=GUI
9012 IFIP=174THENMAI=SSI
9010 PRINTW1,CHR$(27)"L"CHR$(11);CHR$(0)
;:PRINTW1,MAI+CHR$(0)+CHR$(0);:IP$="":GO
TOSB0
9035 IFIP=173THENPRINTW1,CHR$(27)"-CHR
$(1);:IP$=""
9040 IFIP=189THENPRINTW1,CHR$(27)"E";:I
P$=""
9045 IFIP=188THENPRINTW1,CHR$(27)"E"CHR
$(27)"-CHR$(1);:IP$=""
9050 IFIP=190THENIP#=CHR$(14)
9055 IFIP=191THENPRINTW1,CHR$(27)"-CHR
$(1);:IP#=CHR$(14)
9060 IFIP=170THENPRINTW1,CHR$(27)"E";:I
F#=CHR$(14)
9065 IFIP=167THENPRINTW1,CHR$(27)"E"CHR
$(27)"-CHR$(1);:IP#=CHR$(14)
9067 IFIP=164THENPRINTW1,CHR$(27)"F"CHR
$(27)"T"CHR$(27)"H"CHR$(27)"-CHR$(0)...
IP#=CHR$(20)
9070 IFIP=171THENZA=36:IP$=""
9072 IFIP=179THENZA=54:IP$=""
9074 IFIP=177THENZA=72:IP$=""
9076 IFIP=172THENPRINTW1,CHR$(27)"S"CHR
$(0);:IP$=""
9078 IFIP=187THENPRINTW1,CHR$(27)"S"CHR
$(1);:IP$=""
9090 IFZB=7ATHENGOSUB8500
9092 GOTOSB0
9100 INPUTW1,W2,W3,X,Y$:IFVAL(W$)>0TH
ENPRINT"W",W$,X,Y$;"\n"
9110 RETURN
9200 OPEN2,8,2,DB$:GOSUB9100:IFVAL(W$)<>
65THENCLOSE2:PRINTW1,"SO:"*CB$:GOTO6220
9210 IFVAL(W$)=65THENPRINT"-ATEI UEBERSC
HEIDEN \J/N)
9220 GETZ$:IFZ$=""THEN9220
9230 IFZ$="N"THENCLOSE2:CLOSE15:GOTO6200
9240 IFZ$="J"THENCLOSE2:PRINTW1,"SO:"*CZ
$:GOSUB9100:GOTO6220
9250 GOTO9220
10000 X$="":X=0:0=0:W=0:PRINT"\n EXTPOG
RAMM \J/N)
10010 PRINT"VENUE"SPC(34)"-----"
10020 IFTEXT EINGEBEN"SPC(64)
10030 PRINTW1,"32-"EXT KORRIGIEREN"SPC(64)
10040 PRINTW1,"32-"EXT FORMATIEREN"SPC(61)
10050 PRINTW1,"35-"EXT AUSDRUCKEN"SPC(62)
10060 PRINTW1,"36-"ISKETTERVERMALUNG"SPC(58)
10070 PRINTW1,"38-"RIEFPROFF EINGEBEN"SPC(
59)"37-"NOXXXXXXXXXXXXXXX-INCARB
10080 GETX$:X=VAL(X$):IFX<0ORX>7THEN1006
0
10070 IFX=7THENPOKE769,49:PRINT"L":END
10075 IFX=5THEN6000
10080 ONXGOSUB2000,3000,4000,5000,:1000
:GOTO6000
10020 PRINT"GEHEN SIE BITTE DEN TEXT D
ES PRAEFKOPFEINEN :PARTUM = 0)
10010 INPUT"DATEI 1 LI: ";W1$:IFW1$=""OTH
ENRETURN
10015 INPUT"DATEI 2 LI: ";W2$
10020 INPUT"DATEI 3 LI: ";W3$:INPUT"DA
TEI 4 LI: ";W4$
10030 INPUT"TRASSE: ";W4$:INPUT"LAUF/GO
HNORT/76: ";W5$
10050 INPUT"ELEFON: ";W6$:INPUT"ATUM: "
;:W7$
10060 PRINT"UNSCRIFT EINGEBEN? (J/N)
10070 GETY$:IFY$=""THEN10070
10080 IFY$="N"THENRETURN
10090 IFY$="J"THEN1110
1100 GOTOT070
1110 PRINTW1,F0R:P1TOT5:PRINT"DATEI"P
;:INPUTA(P):NEXTP:RETURN

```

READY.

Listing von »Quicktext«

Die Anwendung des

Der hier vorgestellte »elektronische Lehrerkalender« entstand aus dem Bedürfnis heraus, sich die im Lehrerberuf anfallende tägliche Routinearbeit zu erleichtern. Das Programm übernimmt die Verwaltung der Noten einer Schulklasse und stellt einige Funktionen zur Verfügung, mittels derer sich der Lehrer schnell einen Überblick über den Leistungsstand der Klasse verschaffen kann.

Nachdem das Programm geladen wurde, erscheint zunächst das Hauptmenü (Bild 1). Durch Wahl einer der im Menü angezeigten Ziffern können nun die einzelnen Programmteile aufgerufen werden.

Bei der ersten Anwendung wird man sinnvoller Weise den Punkt »1« anwählen, um die Klassenliste anzufertigen. Dabei braucht man sich nur an die Bildschirmanweisungen zu halten. Verläßt man diesen Programmteil, so ist ein Rücksprung dorthin unmöglich, um Fehler in der Notenzuordnung zu vermeiden. Das gilt auch dann, wenn nach dem Programmstart Daten von Disk oder Band geholt wurden. Nach Verlassen des Punktes »1« oder nach Laden der benötigten Daten erscheint unter Punkt »2« die Übersicht über Namen und Noten einzelner Arbeiten sowie die Durchschnittsnote der geschriebenen Arbeiten (Bild 2). Im Programmteil »3« wird den erreichten Punkten die entsprechende Note zugewiesen. Dazu wird zunächst nach der Nummer der Arbeit und der Gesamtpunktzahl gefragt. Anschließend gibt der Computer eine Standardbewertung aus, die man übernehmen oder auch durch eine andere ersetzen kann. Die notwendigen Schritte sind auf dem Schirm erkennbar.

Nach der Notenverteilung wird nach dem Namen und der erreichten Punktzahl ge-

fragt. Die Namen können vollständig oder abgekürzt auf die ersten vier Buchstaben eingegeben werden. Jede Eingabe muß mit »RETURN« abgeschlossen werden. Anschließend erscheint die erreichte Note am rechten Bildschirmrand.

Will man sich den Überblick über Klassenarbeiten ausgeben lassen, so wählt man die »4«. Nach Eingabe der Arbeitsnummer erscheint der entsprechende Klassenspiegel.

Die Programmteile »5« und »6« dienen der Datenpflege. In »5« können Neuzugänge an die Liste angefügt werden. Die Vorgehensweise entspricht dem Teil »1«. In »6« können eingetragene Arbeiten wieder gelöscht werden. Dieser Programmteil wird durch ein Untermenü gesteuert, das auch den Rücksprung ins Hauptmenü gestattet. Alle Arbeiten der Nummer X werden durch die Ziffer »2«, Arbeiten einzelner Schüler durch die »1« gelöscht. Die Eingabe der Schülernamen erfolgt wie im Programmteil »Notenverteilung«.

Sollen alle gespeicherten Daten ausgedruckt werden, so wählt man die »7« (Bild 3). Getestet ist dieser Teil auf dem Nachfolger des VC 1525, dem MPS 801.

Um das Programm zu beenden, drückt man schließlich die »8«. Dieser Teil sorgt für die Datensicherung auf Diskette oder Band.

(Dirk Schwerdfeger)

Bild 3. Auf Knopfdruck möglich — Ausdruck von Klassenspiegeln zu allen Arbeiten

KLASSENSPIEGEL

1 .ARBEIT

I	II	III	IV	V	VI
2	2	2	2	2	2

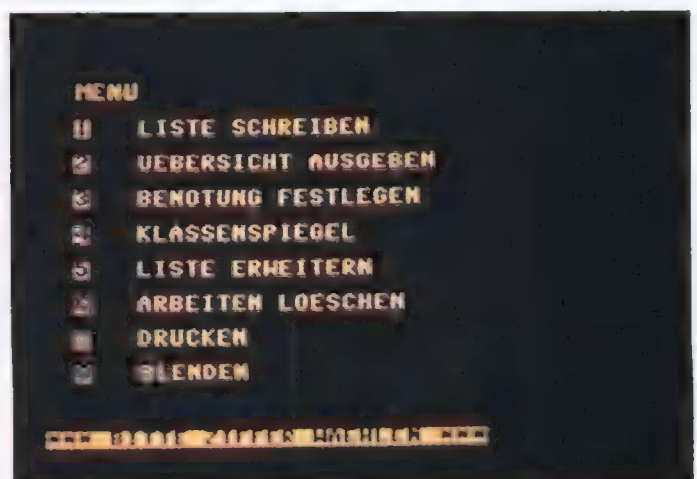
2 .ARBEIT

I	II	III	IV	V	VI
2	2	2	2	2	2

3 .ARBEIT

I	II	III	IV	V	VI
1	1	1	2	1	1

Bild 1. Das Hauptmenü des Lehrerkalenders



**Listing vom
»Lehrerkalender«
(Fortsetzung)**

```

2130 IFZ$="2"THEN500
2999 REM ***NOTENVERTEILUNG***
3000 IFY=0THENGOSUB15000:X2=1
3005 PRINT"0000000000NOTENVERTEILUNG="
3010 INPUT"0000000000ARBEIT NR.: ";T
3020 INPUT"0000GESAMTPUNKTZAHL: ";GP
3025 REM ***VERT. BERECHNEN***
3030 PRINT"03STANDARDVERTEILUNG:="
3040 Z1=INT(GP*.98+.5):Z2=INT(GP*.85+.5)
:Z3=INT(GP*.7+.5):Z4=INT(GP*.5+.5)
3050 Z5=INT(GP*.25+.5)
3060 PRINT"0SEHR GUT:"GP"-Z1
3070 PRINT"01"GUT:"Z1-1"-Z2
3080 PRINT"02BEFRIED.:"Z2-1"-Z3
3090 PRINT"03AUSREI.:"Z3-1"-Z4
3100 PRINT"04MANGELH.:"Z4-1"-Z5
3110 PRINT"05JUNGENUE.:"Z5-1"-0"
3120 PRINT"03MOECHTEN SIE EINE ANDERE WE
RTUNG?(J/N)="
3130 GETZ$:IFZ$=""THEN3130
3140 IFZ$="N"THENPRINT"03DIESE WERTUNG W
ERDE ICH ANWENDEN "="GOTO3145
3142 GOTO3150
3145 FORI=0TO3000:NEXT:GOTO3300
3150 IFZ$="J"THEN3170
3160 GOTO3130
3165 REM ***EIGENWERTUNG EINGEBEN***
3170 PRINT"Sooooooooooooo"
IIIIEIGENWERTUNG:?"
3180 PRINT"00000000000000000000"GP="-";
:INPUTZ1
3190 PRINT"00000000000000000000"Z1-1"-
";:INPUTZ2
3200 PRINT"00000000000000000000"Z2-1"-
";:INPUTZ3
3210 PRINT"00000000000000000000"Z3-1"-
";:INPUTZ4
3220 PRINT"00000000000000000000"Z4-1"-
";:INPUTZ5
3230 PRINT"00000000000000000000"Z5-1"-
0"
3240 PRINT"03IHRE WERTUNG WERDE ICH ANWE
NDEN ="
3250 FORI=0TO3000:NEXT
3299 REM ***NOTEN BERECHNUNG***
3300 PRINT"0000ARBEIT NR.: ";T
3310 PRINT"0000GES-PUNKTE : ";GP
3320 PRINT"000000NAME: 0000FERR. PUNKTE: 0000
IIIINOTE: "
3340 PRINT
3350 K=K+1:INPUTB$:IFB$=""OR $K=N$ THEN500
3360 INPUT"00000000000000000000";EP
3370 IFEP>=Z1THENEN=1
3380 IFEP>=Z2ANDEP<Z1THENEN=2
3390 IFEP>=Z3ANDEP<Z2THENEN=3
3400 IFEP>=Z4ANDEP<Z3THENEN=4
3410 IFEP>=Z5ANDEP<Z4THENEN=5
3420 IFEP<Z5THENEN=6
3499 REM ***NAMEN-SUCH-ROUTINE***
3500 FORJ=1TON
3510 IFLEN(B$)=4THEN3530
3511 L1=LEN(B$):IFL1=10THEN3520
3512 L2=10-L1:FORO=1TOL2:B$=B$+" ":NEXTO
3520 IFB$=A$(J) THENM=J:A(M,T)=EN:J=N:GOT
O3550
3530 IFLEFT$(A$(J),4)=B$THENM=J:A(M,T)=E
N:J=N:GOTO3550
3540 IFJ=N THENPRINT"0000000000NICHT G
EFUNDEN
":XY=1
3550 NEXTJ
3555 IFXY=1THEN3610

```



```

3560 PRINT"#####
#####A(M,T)
3610 XY=0:GOTO3350
3999 REM ***KLASSENSPIEGEL***
4000 IFY=0THENGOSUB15000:X2=1
4005 PRINT"#####KLASSENSPIEG
EL"
4010 PRINT"#####DEN KLASSENSPIEGEL WELCHER
ARBEIT "
4020 PRINT"#####WOLLEN SIE SEHEN? (1-6)"
4030 GETZ$:IFZ$=""THEN4030
4040 IFASC(Z$)<49ORASC(Z$)>54THEN4030
4050 Z=VAL(Z$)
4210 PRINT"#####Z"II. ARBEI
T"
4250 PRINT"#####I III IIII IV V VI
"
4260 PRINT"#####
"
4270 PRINT"#####
"
4280 J=Z:GOSUB17000
4299 REM **VERTEILUNG AUSGEBEN**
4300 PRINT"#####TAB(6)KS
4310 PRINT"#####TAB(10)KG
4320 PRINT"#####TAB(14)KB
4330 PRINT"#####TAB(18)KA
4340 PRINT"#####TAB(22)KM
4350 PRINT"#####TAB(26)KU
4400 PRINT"#####WEITER MIT '1' / ZURUECK
MIT '2'"
4410 GETX$:IFX$=""THEN4410
4420 IFASC(X$)<49ORASC(X$)>50THEN4410
4430 IFX$="2"THEN500
4440 IFX$="1"THEN4000
4999 REM ***LISTE ERWEITERN***
5000 IFY=0THENGOSUB15000:X2=1
5010 PRINT"#####FACH: ";FA$,"KLASSE: ";KL$
5020 PRINT"#####HALBJ-BEGINN: ";SJ$
5030 PRINT"#####BITTE JETZT NEUE NAMEN EIN
GEBEN."
5040 PRINT"#####(MAX. LAENGE : 10 ZEICHEN)"
5050 PRINT"#####*' SCHLIESST DIE LISTE."
5060 N=N-1
5070 N=N+1
5080 INPUT"NAME";A$(N)
5090 IFLEN(A$(N))>10THENPRINT"ZUVIELE ZE
ICHEN!":GOTO5080
5100 IFA$(N)="*"THEN5150
5110 L1=LEN(A$(N)):IFL1=10THEN5140
5120 L2=10-L1:FORJ=1TOL2:A$(N)=A$(N)+" "
:NEXTJ
5140 GOTO5070
5150 PRINT"#####LISTE GESCHLOSSEN#####
#####ZURUECK MIT EINER TASTE":Y=1
5160 GETZ$:IFZ$=""THEN5160
5170 GOTO500
5999 REM ***ARBEITEN LOESCHEN***
6000 IFY=0THENGOSUB15000:X2=1
6010 PRINT"#####WELCHE ARBEIT SOLL GELOESCHT
WERDEN?"
6020 PRINT:INPUT"(1-6) ";W
6030 PRINT"#####EINZELNE ARBEITEN"
6035 PRINT"#####ALLE ARBEITEN DER NR.
:";W
6040 PRINT"#####ZURUECK INS MENU"
6050 GETZ$:IFZ$=""THEN6050
6060 IFASC(Z$)<49ORASC(Z$)>51THEN6050
6070 Z=VAL(Z$):ONZGOTO6100,6500,500
6099 REM ***SCHUELERA. LOESCHEN***
6100 PRINT"#####BITTE NAMEN DES SCHUELERS EI

```

```

NGEBEN,"
6110 PRINT"#####DESSEN ARBEIT GELOESCHT WERD
EN SOLL."
6120 PRINT"#####":INPUT"NAME";B$
6130 FORJ=1TON
6140 IFLEN(B$)=4THEN6180
6150 L1=LEN(B$):IFL1=10THEN6170
6160 L2=10-L1:FORO=1TOL2:B$=B$+" ":NEXTO
6170 IFB$=A$(J)THENM=J:A(M,W)=0:GOTO6190
6180 IFLEFT$(A$(J),4)=B$THENM=J:A(M,W)=0
6190 NEXTJ
6200 PRINT"#####DIE ARBEIT VON "A$(M)
6210 PRINT"#####WURDE GELOESCHT.":FORU=0T099
9:NEXTU:GOTO500
6499 REM ***ARB.-NR.X LOESCHEN***
6500 PRINT"#####DIE ARBEIT NR.: "W" WIRD GELO
ESCHT."
6510 FORI=1TON
6520 A(I,W)=0
6530 NEXTI
6540 FORU=1T0500:NEXTU
6550 GOTO500
6999 REM ***AUSDRUCK***
7000 IFY=0THENGOSUB15000:X2=1
7010 PRINT"#####DIE KLASSENLISTE DER"KL
$"WIRD"
7020 PRINT"#####NUN AUSGEDRUCKT."
7030 PRINT"#####IST DER DRUCKER BEREIT ?
(J/N)"
7040 GETQ$:IFQ$=""THEN7040
7050 IFQ$<>"J"THEN500
7060 PRINT"#####BITTE ETWAS GEDULD.
"
7065 REM ***DRUCK-ROUTINE***
7070 OPEN4,4,0
7075 PRINT#4,"KLASSE: "KL$,"FACH: "FA$
7077 PRINT#4,"HALBJ-BEGINN: "SJ$
7080 PRINT#4,"
"
7090 FORI=1TON-1:GOSUB16000
7100 PRINT#4,A$(I)" "A(I,1)" "A(I,2)" "A
(I,3)" "A(I,4)" "A(I,5)" "A(I,6)" "A(I,7
)
7110 NEXTI
7120 PRINT#4,"
"
7200 CLOSE4
7500 PRINT"#####SOLLEN DIE KLASSENSPIEGEL AU
CH "
7600 PRINT"#####AUSGEDRUCKT WERDEN ? (J/N)"
7610 GETQ$:IFQ$=""THEN7610
7620 IFQ$="N"THEN500
7630 IFQ$="J"THEN7700
7640 GOTO7610
7700 PRINT"#####NUN WERDEN DIE KLASSENS
PIEGEL"
7710 PRINT"#####AUSGEDRUCKT.":F=6
7800 OPEN4,4
7805 PRINT#4,"KLASSENSPIEGEL"
7810 FORJ=1TOF:GOSUB17000
7820 IFKS=0ANDKG=0ANDKB=0ANDKA=0ANDKM=0A
NDKU=0THENF=J:GOTO7890
7830 PRINT#4,J".ARBEIT"
7840 PRINT#4,"#####I II III IV I
V VI "
7850 PRINT#4,"#####
"
7860 PRINT#4,KS;TAB(2)KG;TAB(2)KB;TAB(2)
KA;TAB(2)KM;TAB(2)KU
7890 NEXTJ
7900 CLOSE4:GOTO500

```

Listing vom »Lehrerkalender«
(Fortsetzung)


```

9999 REM ***BEENDEN***
10000 PRINT"ZUNACHST BEVOR DAS PROGRAMM ENDE
T,MUESSEN"
10010 PRINT"ZUNACHST ALLE DATEN ABGES
PEICHERT"
10020 PRINT"WERDEN."
10030 PRINT"1. AUF KASSETTE SPEIC
HERN"
10040 PRINT"2. AUF DISKETTE SPEICHE
RN"
10045 PRINT"3. BITTE ZIFFER WAEHL
EN"
10050 GETZ$:IFZ$=""THEN10050
10060 IFASC(Z$)<49ORASC(Z$)>50THEN10050
10070 Z=VAL(Z$):ONZGOTO10100,11000
10099 REM ***DATA-SAVE / TAPE***
10100 PRINT"1. BITTE KASSETTE AUF FREIE
N "
10110 PRINT"2. SPEICHERPLATZ VORSPULEN."
10120 PRINT"3. HABEN SIE DAS BAND POSIT
IONIERT? (J/N)"
10130 GETZ$:IFZ$=""THEN10130
10140 IFZ$="J"THEN10200
10150 IFZ$="N"THEN10100
10160 IFZ$<>"J"ANDZ$<>"N"THEN10130
10200 PRINT"4. DIE DATEN WERDEN NUN GESP
EICHERT."
10210 PRINT"5. ETWAS GEDULD BITTE.":FI$=K
L$+FA$
10220 OPEN1.1.1.FI$
10230 PRINT#1,KL$
10240 PRINT#1,FA$
10250 PRINT#1,SJ$
10260 FORI=1TON
10270 PRINT#1,A$(I)
10280 NEXTI
10290 FORI=1TON
10300 FORJ=1TO6
10310 PRINT#1,A(I,J)
10320 NEXTJ
10330 NEXTI
10415 PRINT#1,Y
10420 CLOSE1:GOTO12000
10999 REM ***DATA-SAVE / DISK***
11000 PRINT"6. DIE DATEN WERDEN NUN GESP
EICHERT."
11010 PRINT"7. ETWAS GEDULD BITTE.":FI$="
S:"+KL$+FA$+",S,W"
11020 OPEN2,8,2,FI$
11030 PRINT#2,KL$
11040 PRINT#2,FA$
11050 PRINT#2,SJ$
11060 FORI=1TON
11070 PRINT#2,A$(I)
11080 NEXTI
11090 FORI=1TON
11100 FORJ=1TO6
11110 PRINT#2,A(I,J)
11120 NEXTJ
11130 NEXTI
11215 PRINT#2,Y
11220 CLOSE2:GOTO12000
11999 REM ***SCHLUSSBILD***
12000 PRINT"8. SCHLUSSBILD"
12010 PRINT"9. SCHUESS!"
12020 PRINT"10. IHR BO
ZLO VON SCHWERFG":END
14999 REM ***DATEN LADEN***
15000 PRINT"11. DAS PROGRAMM BEARBEITEN
ZU KOENNEN,"
15010 PRINT"12. MUESSEN ZUVOR DIE DATEN EIN

```

```

GELESEN"
15020 PRINT"13. WERDEN."
15022 INPUT"14. KLASSE: ";KL$
15024 INPUT"15. FACH : ";FA$
15026 FI$=KL$+FA$
15030 PRINT"16. 1. VON KASSETTE"
15040 PRINT"17. 2. VON DISKETTE"
15050 PRINT"18. BITTE ZIFFER WA
EHLEN"
15060 GETZ$:IFZ$=""THEN15060
15070 IFASC(Z$)<49ORASC(Z$)>50THEN15060
15075 PRINT"19. DIE DATEN WERDEN EING
ELESEN."
15077 PRINT"20. BITTE ETWAS GEDULD."
15080 Z=VAL(Z$):ONZGOTO15200,15600
15099 REM ***DATA-LOAD / TAPE***
15200 OPEN1,1,0,FI$
15210 INPUT#1,KL$
15220 INPUT#1,FA$
15230 INPUT#1,SJ$
15240 N=N+1
15250 INPUT#1,A$:A$(N)=A$
15260 IFA$(N)="*"THEN15280
15270 GOTO15240
15280 FORI=1TON
15290 FORJ=1TO6
15300 INPUT#1,A(I,J)
15310 NEXTJ:NEXTI
15400 INPUT#1,Y
15410 CLOSE1
15500 RETURN
15599 REM ***DATA-LOAD / DISK***
15600 FI$=FI$+",S,R": OPEN2,8,2,FI$
15610 INPUT#2,KL$
15620 INPUT#2,FA$
15630 INPUT#2,SJ$
15640 N=N+1
15650 INPUT#2,A$:A$(N)=A$
15660 IFA$(N)="*"THEN15680
15670 GOTO15640
15680 FORI=1TON
15690 FORJ=1TO6
15700 INPUT#2,A(I,J)
15710 NEXTJ:NEXTI
15800 INPUT#2,Y
15810 CLOSE2
15900 RETURN
15999 REM ***DURCHSCHNITT BERECHNEN***
16000 DN=0:AZ=0
16010 FORJ=1TO6
16020 DN=DN+A(I,J)
16030 IFA(I,J)=0THEN16050
16040 AZ=AZ+1
16050 IFJ=6ANDAZ=0THENAZ=1:REM **DIV.DUR
CH 0 VERHINDERN**
16060 NEXTJ
16070 A(I,7)=INT((DN/AZ)*10+.5)/10
16080 RETURN
16999 REM ***KL.-SPIEGEL BERECHNEN***
17000 KS=0:KG=0:KB=0:KA=0:KM=0:KU=0
17010 FORI=1TON
17020 IFA(I,J)=1THENKS=KS+1
17030 IFA(I,J)=2THENKG=KG+1
17040 IFA(I,J)=3THENKB=KB+1
17050 IFA(I,J)=4THENKA=KA+1
17060 IFA(I,J)=5THENKM=KM+1
17070 IFA(I,J)=6THENKU=KU+1
17080 NEXTI
17090 RETURN

```

Listing vom «Lehrerkalender»
(Schluß)

READY.

Supervoc

Supervoc ist ein Lernprogramm mit dem man einfach und unkompliziert Vokabeln lernen kann. Das Abhören besteht aus drei Runden, die sich durch den Abhörmodus unterscheiden und die man jeweils erreicht, wenn man in einer Runde weniger als zwei Fehler gemacht hat. Es ist möglich, Vokabeln von Kassette zu laden oder per Tastatur bis zu 50 Vokabeln einzugeben.

Nach dem Programmstart erscheint zunächst das Titelbild, dann die Frage: »Vokabeln von Kassette (j/n)?«. Wird die Frage mit »j« beantwortet, kann man ein Vokabelfile von Kassette laden (sofern bereits eines vorhanden ist). Bei »n« springt das Programm zur Routine »Vokabeln eingeben«. Hier kann man jeweils eine Vokabel und die drei Bedeutungen der Vokabel eingeben. Anschließend gelangt man zum Menü (Bild 1).

Das Menü

Hier kann man wählen zwischen:

- j (Abhören starten)
- c (Vokabeln verbessern)
- l (Neue Vokabeln)
- s (Vokabeln speichern)

Abhören

Oben links erscheint die Vokabel, deren Bedeutung

man eine Zeile darunter eingeben kann (Bild 2). Eine eventuell falsch eingegebene Bedeutung kann durch »Home« wieder gelöscht werden.

Leichter lernen

Nach »Return« überprüft das Programm die Antwort auf ihre Richtigkeit und zeigt das Ergebnis an.

Bei falschem Ergebnis werden alle richtigen Bedeutungen angezeigt.

Nachdem alle Vokabeln einmal an der Reihe waren, erscheint die Bewertung und eventuell erfolgt ein Sprung zur nächsten Runde (bei weniger als zwei Fehlern).

Verbessern

Mit der Taste n kann man aus dem Vokabelspeicher eine Vokabel auswählen und

dann mit der Taste c dem Computer mitteilen, daß man sie verbessern möchte.

Danach erscheint zunächst die Vokabel, dann die Bedeutungen. Jetzt kann man bei Bedarf eine Form neu eingeben. Nach der 50. Vokabel erfolgt automatisch ein Sprung zum Menü.

Laden und Speichern

Das Laden und Speichern von Vokabeln ist auf den Betrieb mit dem Kassettenlaufwerk abgestimmt. Durch Ändern der Geräteadresse bei den entsprechenden OPEN-Statements ist aber auch das

Arbeiten mit Floppy-Laufwerken möglich. Eine Besonderheit des Programms ist die Tatsache, daß innerhalb einer Runde keine Vokabel doppelt erscheint.

Schneller lernen

Dennoch werden die Vokabeln durcheinander abgefragt, und die maximale Zugriffszeit für eine Vokabel beträgt bei vollem Speicher nur 0,1 Sekunden.

(Michael Schmidt)

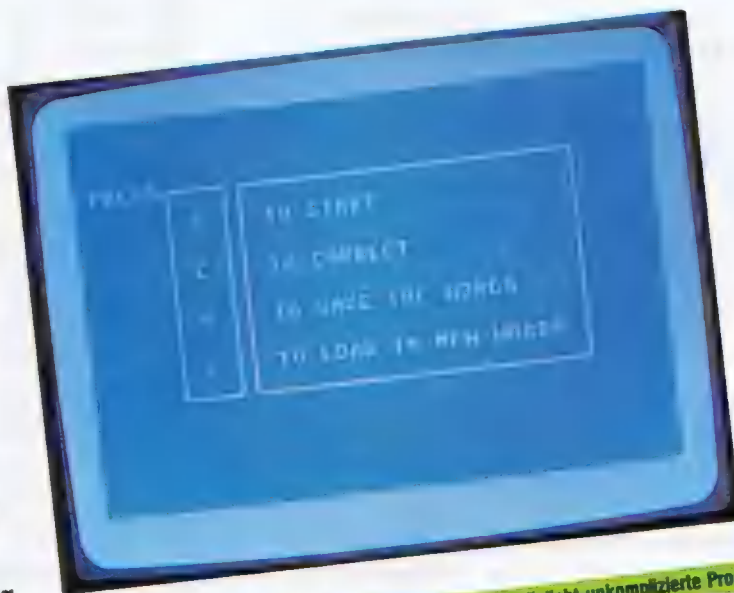


Bild 1. Das Supervoc-Menü ermöglicht unkomplizierte Programmbedienung

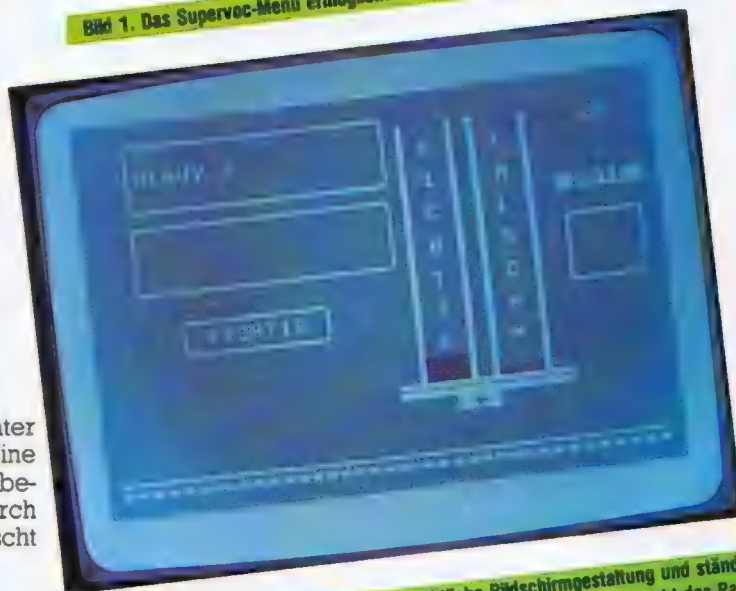


Bild 2. Supervoc fragt Vokabeln ab. Übersichtliche Bildschirmgestaltung und ständige Übersicht über den Lernfortschritt durch zwei Säulendiagramme - so macht das Lernen Spaß


```

100 REM *****
110 REM * MICHAEL SCHMIDT * 1984 MSC.*
120 REM * B A D I S C H E 6 4 ' E R *
130 REM * MSC *
140 REM * SUPERVOC. * COMMODORE 64*
150 REM *****
160 :
170 :
180 GOSUB2060:POKE198,0:WAIT198,1
190 POKE32768,96:POKE650,234
200 DIMV$(50,4):PF=1732:PR=1727:SY=1
210 PRINT"┐":PRINT:PRINT:PRINT
220 PRINT" VOKABELN VON KASSETTE ( J /
N ) ?"
230 GETA$: IFA$="J" THEN GOSUB1320:GOSUB157
0:GOTO750
240 IFA$="N" THEN GOSUB1630:GOSUB1570:GOTO
750
250 GOTO230
260 GOSUB1060
270 N=N(RND(TI)*9):R=0:F=0:PR=1727:PF=17
32
280 M=0
290 Z=Z+1:M=M+N:IFM>50 THEN M=M-50
300 IFM=N THEN M1=M+1:IFM1=2 THEN M1=0:Z=0:
GOTO720
310 POKE214,2:PRINT:PRINTTAB(4)"┐
"
320 POKE214,1:PRINT:PRINTTAB(34)Z
330 IFV$(M,1)="*" THEN 290
340 IFSX=1 THEN SY=INT(RND(TI)*2+1)
350 POKE214,7:PRINT:PRINTTAB(4)"
"
360 POKE214,2:PRINT:PRINTTAB(4)V$(M,SY):
" ?"
370 POKE214,7:PRINT:B$=""
380 FORI=1TO15
390 POKE198,0:WAIT198,1:GETA$
400 IFA$="B" THEN POKE214,7:PRINT:PRINTTAB
(4)"┐":B$="":GOTO380
410 IFASC(A$)=13 THEN 460
420 IFA$="┐" OR A$="┐" OR A$="┐" OR A$="┐" OR A$
="┐" OR A$="┐" OR A$="┐" THEN 390
430 PRINTTAB(4)A$:
440 B$=B$+A$
450 NEXT
460 IFSY=2 THEN 490
470 FORJ=2TO4:IFV$(M,J)<>B$ OR B$="" THEN NE
XT:GOTO510
480 GOTO630
490 IFV$(M,1)=B$ THEN 630
500 GOTO510
510 FZ=FZ+1
520 POKE214,12:PRINT:PRINTTAB(7)"┐ FALSC
H* "
530 FORJ=1TO255:NEXT:POKE214,12:PRINT:PR
INTTAB(7)"┐ FALSCH* "
540 PRINT"┐"
550 F=F+1:IFF>3 THEN F=1:PF=PF-40
560 FORI=PFTOPF+2:POKEI,F(F):POKE54272+I
,0:NEXT
570 POKE214,20:PRINT"┐":MM=0
580 FORI=3TO5:PRINT" : ";:FORJ=1TOLEN(V$
(M,I-SY))
590 MM=MM+1
600 PRINTMID$(V$(M,I-SY),J,1);
610 FORK=1TO100:NEXT:NEXT:NEXT
620 FORI=MMTO31:PRINT" ";:NEXT:PRINT:GOT
O290
630 RZ=RZ+1
640 POKE214,12:PRINT:PRINTTAB(7)"┐ RICHT
IG "
650 FORK=1TO255:NEXT:POKE214,12:PRINT:PR
INTTAB(7)"┐ RICHTIG "
660 PRINT"┐"
670 R=R+1:IFR>3 THEN R=1:PR=PR-40
680 FORI=PRTOPR+2:POKEI,R(R):POKE54272+I
,0:NEXT
690 POKE214,20:PRINT:PRINT"
";
700 PRINT"┐ _____
":GOTO290
710 GOTO290
720 PRINT"┐"
730 IFFZ<2 THEN 820
740 GOSUB870
750 RZ=0:FZ=0
760 GOSUB1940
770 GETA$: IFA$="Y" THEN GOSUB1060:GOTO270
780 IFA$="S" THEN GOSUB1430:GOTO750
790 IFA$="C" THEN GOSUB1770:GOTO750
800 IFA$="L" THEN RUN200
810 GOTO770
820 IFSY=2 AND SZ=0 THEN SZ=1:GOSUB970:SY=1:
GOTO750
830 IF SZ=1 THEN 1010
840 GOSUB930
850 SY=2
860 GOTO760
870 PRINT"┐┐┐":PRINT" VON "FZ+RZ" HABEN
SIE┐┐"
880 PRINTTAB(10)RZ"RICHTIG UND┐┐"
890 PRINTTAB(10)FZ"FALSCH ┐┐"
900 PRINT"ICH SCHLAGE IHNEN VOR DIE VOKA
BELN┐"
910 PRINT"NOCH EINMAL ZU WIEDERHOLEN"
920 FORI=1984TO2023:POKEI,160:POKE54272+
I,14:FORJ=1TO100:NEXT:NEXT:RETURN
930 PRINT"┐┐┐ BRAVO!┐┐":PRINT"SIE HABE
N VON"FZ+RZ"VOKABELN┐"
940 PRINTTAB(10)RZ"GEWUSST┐┐"
950 PRINT"ICH HORE SIE JETZT UMGEGEHRT A
B"
960 FORI=1984TO2023:POKEI,160:POKE54272+
I,14:FORJ=1TO100:NEXT:NEXT:RETURN
970 PRINT"┐┐┐ DRITTE RUNDE┐┐"
980 PRINTRZ"RICHTIGE "FZ" FALSCH ┐┐"
990 PRINT"UND JETZT GEHT'S DURCHEINANDER
"
1000 FORI=1984TO2023:POKEI,160:POKE54272
+I,14:FORJ=1TO100:NEXT:NEXT:RETURN
1010 PRINT"┐┐┐ SO, DAS DUERFTE GENUEGEN!
┐┐"
1020 PRINT"WIEDERHOLEN SIE DIESE VOKABEL
N ┐┐"
1030 PRINT"IN CA 3-4 WOCHEN┐┐"
1040 PRINTN$" VERABSCHIEDET SICH !
1050 FORI=1TO4000:NEXT:RUN
1060 PRINT"┐"
1070 FORI=1TO2
1080 PRINT" _____ "
1090 FORJ=1TO3
1100 PRINT" | | "
1110 NEXT
1120 PRINT" _____ "
1130 NEXT
1140 PRINT:PRINTTAB(6)" _____ "
1150 PRINTTAB(6)" | | "
1160 PRINTTAB(6)" _____ "

```



```

1170 POKE214,21:PRINT:PRINT"
*****
1180 PRINT" *****
*****
1190 POKE214,1:PRINT
1200 FORI=1TO16:PRINTTAB(22)"  I  "
1210 NEXT
1220 PRINTTAB(21)"-----"
1230 PRINTTAB(26-LEN(N$)/2)"  ";:FORI=1T
010:PRINTMID$(N$,I,1);:NEXT:PRINT"  "
1240 POKE214,2:PRINT
1250 FORI=1TO7:PRINTTAB(24)R$(I);TAB(29)
F$(I):PRINT:NEXT
1260 POKE214,5:PRINT
1270 PRINTTAB(33)"  NOTE  "
1280 PRINT:PRINTTAB(33)"
1290 FORI=1TO3:PRINTTAB(33)"  I  ":NEXT
1300 PRINTTAB(33)"
1310 RETURN
1320 PRINT"*****";:POKE32768,76
1330 OPEN1,1,0,"VOKABELN"
1340 INPUT#1,N$
1350 FORI=1TO50
1360 INPUT#1,V$(I,1)
1370 INPUT#1,V$(I,2)
1380 INPUT#1,V$(I,3)
1390 INPUT#1,V$(I,4)
1400 PRINTI:NEXT
1410 CLOSE1
1420 RETURN
:
1430 SYS32768:PRINT"  "
1440 INPUT"NAME DES PROGRAMMS
*****";N$:PRINT
1450 N$=LEFT$(N$,10)
1460 PRINT"*****" N$ "  "
1470 OPEN1,1,1,"VOKABELN"
1480 PRINT#1,N$
1490 FORI=1TO50
1500 PRINT#1,V$(I,1)
1510 PRINT#1,V$(I,2)
1520 PRINT#1,V$(I,3)
1530 PRINT#1,V$(I,4)
1540 NEXT
1550 CLOSE1
1560 RETURN
:
1570 REM
1580 FORI=1TO7:READJ$,K$:R$(I)=J$:F$(I)=
K$:NEXT
1590 FORI=0TO9:READJ:N(I)=J:NEXT
1600 FORI=1TO3:READJ:F(I)=J:NEXT
1610 FORI=1TO3:READJ:R(I)=J:NEXT
1620 RETURN
1630 FORI=1TO50
1640 PRINT"  "
1650 PRINT:PRINTTAB(5)"  EINGEBEN NEUER
VOKABELN ! "I"  "
1660 POKE214,4:PRINT:PRINT
1670 INPUT"VOKABEL :";V$(I,1)
1680 IFV$(I,1)=""THENGO SUB1890:RETURN
1690 IFLEN(V$(I,1))>15THEN1660
1700 FORJ=2TO4
1710 POKE214,2+3*J:PRINT
"
1720 INPUT"BEDEUTUNG :";V$(I,J)
1730 IFV$(I,J)=""THENV$(I,J)="*"
1740 IFLEN(V$(I,J))>15THEN1710
1750 NEXT:PRINT
1760 RETURN
1770 FORI=1TO50
1780 PRINT"  ":PRINT:PRINTTAB(5)"  VERBES
SERN VON VOKABELN ! "I"  "

```

```

1790 PRINT:PRINT" (  NEXT OR  C
CORRECT )
1800 POKE214,6:PRINT:PRINTTAB(10)V$(I,1)
1810 GETA$:IFA$=""THEN1810
1820 IFA$="N"THENNEXT:RETURN
1830 IFA$<>"C"THEN1810
1840 POKE214,11:PRINT
1850 FORJ=1TO4:PRINTV$(I,J);:INPUT"V: ";
V$(I,J)
1860 NEXT:NEXT
1870 RETURN
1880 RETURN
1890 PRINT"***** A LITTLE MOMENT, PL
EASE !"
1900 FORK=1TO50:FORJ=1TO4:V$(K,J)="*":NE
XT:NEXT:RETURN
1910 DATAR,F,I,A,C,L,H,S,T,C,I,H,G,*
1920 DATA3,7,9,13,17,19,21,47,41,37
1930 DATA121,248,160,121,248,160
1940 PRINT"*****"
1950 PRINT" PRESS
1960 FORI=1TO12
1970 PRINT"  I  "
1980 NEXT
1990 PRINT"
2000 POKE214,5:PRINT
2010 PRINTTAB(8)"Y"TAB(13)"TO START"
2020 PRINTTAB(8)"  "TAB(13)"TO CORRECT"
2030 PRINTTAB(8)"  "TAB(13)"TO SAVE THE
WORDS"
2040 PRINTTAB(8)"  "TAB(13)"TO LOAD IN
NEW WORDS"
2050 RETURN
2060 PRINT"*****"
2070 V$="COPYRIGHT BY MICHAEL SCHMIDT : "
2080 PRINTTAB(4);:FORI=1TOLEN(V$)
2090 PRINTMID$(V$,I,1);:FORJ=1TO100:NEXT
:NEXT:PRINT
2100 FORI=1TO500:NEXT
2110 PRINT"  "
2120 PRINT"  I  "
2130 PRINT"  I  "
2140 PRINT"  I  "
2150 PRINT"  I  "
2160 PRINT"  I  "
2170 PRINT"  I  "
2180 PRINT"  I  "
2190 PRINT"  I  "
2200 PRINT"  I  BADISCHE 64'E
R"
2210 PRINT"  "
2220 PRINT"
2230 PRINT"  I  *****
*****  "
2240 PRINT"
2250 PRINT"***** PRESS ANY KEY TO
START ***"
2260 RETURN
READY

```

Listing zu »Supervoc«

KURZ—L A N G—KURZ—

kein Problem

Dieses Programm eignet sich in hervorragender Weise, um das Morsen zu lernen oder zu trainieren.

Dem Anfänger wird die Möglichkeit geboten, durch eine Morsezeichenübersicht die Morsezeichen zu erlernen. Später kann man durch den Programmteil »Computer morst Zufallstext« die erworbenen Kenntnisse überprüfen, wobei die Morsegeschwindigkeit frei gewählt werden kann. Mit etwas Übung gelingt es schon recht bald, zumindest bei niedriger Morsegeschwindigkeit, die meisten Zeichen zweifelsfrei zu identifizieren. Man kann den Computer aber auch dazu bringen, einen eingegebenen Text in Morsezeichen zu übersetzen, was ebenfalls als Trainingsmöglichkeit genutzt werden kann.

Für Profis gibt es dann im dritten Teil die Möglichkeit, selbst gemorsten Text vom Computer überprüfen zu lassen. Als Eingabetaste dient entweder »f7« oder der Feuerknopf beim Joystick.

Gerade dieser Teil des Programms war schwierig zu realisieren, weil die Verarbeitung und Übersetzung in Basic viel zu langsam ist. Letztendlich konnte diese Schwierigkeit aber umgangen werden, indem die gemorste Zeichenfolge nicht unmittelbar ausgewertet, sondern zunächst in einen freien RAM-Bereich zwischengespeichert wird. Erst nachdem der Morsetext zu Ende eingegeben ist, wird dieser Speicherbereich ausgelesen und übersetzt.

Alle Funktionen werden über ein Menü angewählt, wodurch das Programm sehr einfach zu bedienen ist.

(Stefan Ingenhorst)

READY.

```

0 REM *****
1 REM *
2 REM * MORSETRAINER + DOLMETCHER *
3 REM * C BY JENS MARTENS UND
4 REM * STEFAN INGENHORST
5 REM *
6 REM *****
7 REM *****ANDROMAT SOFTWARE*****
8 REM *****
9 GOSUB20000
10 POKE55,200:POKE56,47:CLR:M$="":FORI=5
4272TO54300:POKEI,0:NEXT
20 REM *****
25 REM * NEUE ZEICHEN;77-KURZ,78-LANG *
30 REM *****
40 POKE56334,PEEK(56334)AND254:POKE1,PEE
K(1)AND251
50 FORI=0TO2048:IFI=1ITHENGOSUB10000
60 POKE12288+I,PEEK(53248+I):NEXT
70 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334
)OR1
80 POKE53272,(PEEK(53272)AND240)OR12
90 READA:IFA=-1THEN160
100 FORJ=0TO7:READB:POKE12288+A*8+J,B:NE
XT:GOTO90
110 DATA 77,0,0,0,24,24,0,0,0
120 DATA 78,0,0,0,126,126,0,0,0
130 DATA-1
140 REM *****
150 REM ** SET SOUND **
155 REM *****
160 W=54276:POKE54277,32:POKE54278,242
170 POKE54272,157:POKE54273,69
180 POKE54296,15
190 REM *****
200 REM ** MORSEDATEN LESEN **
201 REM *****
205 DIMM$(255)
210 READM:IFM=-1THEN300
220 READM$:M$(M)=M$:GOTO210
290 REM *****
291 REM ** MENUE **
292 REM *****
300 PRINT"TAB(13)"[CUR* M E N U E *"]
320 PRINT"COMPUTER MORST ZUFA
LLTEXT"
330 PRINT"COMPUTER MORST EINGEGE
BENEN TEXT"

```

Listing zum »Morse-
trainer«


```

340 PRINT"BITTE 3. COMPUTER EMPFANGT"
350 PRINT"BITTE 4. MORSEZEICHEN AUFLISTEN"
360 PRINT"BITTE 5. PROGRAMM BEENDEN"
370 INPUT"BITTE WAELHEN ";M:IFM<1OR M>5THEN300
380 ONMGO TO1000,2000,3000,4000,5000
990 REM *****
991 REM ** ZUFALLSTEXTGENERATION **
992 REM *****
1000 PRINT"BITTE GESCHWINDIGKEIT WAELHEN"
1010 INPUT"(IN WORTEN PRO MINUTE) ";G
1020 M$="":FORI=1TO40:M$=M$+" ":FORJ=1TO 5
1030 A=INT(65*RND(1))+34:IFM$(A)=""THEN1030
1040 M$=M$+CHR$(A):NEXTJ,I
1050 PRINT:PRINT:PRINT
1060 FORM=1TO252STEP36:PRINTMID$(M$,M,36):NEXT
1070 GOTO2500
1997 REM *****
1998 REM ** EIGENER TEXT **
1999 REM *****
2000 PRINT"BITTE TEXT EINGEBEN !"
2020 PRINT"WENN FERTIG, 'F1' DRUECKE N ."
2030 M$=""
2040 GETZ$:IFZ$=""THEN2040
2050 IFASC(Z$)=133THEN2100
2060 IFASC(Z$)=20THENM$=LEFT$(M$,LEN(M$)-1):GOTO2080
2065 IFM$(ASC(Z$))=""THEN2040
2070 M$=M$+Z$:IFLEN(M$)>254THEN2100
2080 PRINT"BITTE M$"
2090 GOTO2040
2100 PRINT"S":FORK=1TO8:PRINTCHR$(17):NEXT:PRINT"TEXT UEBERNOMMEN ."
2110 PRINT"WELCHE GESCHWINDIGKEIT"
2120 INPUT"(IN WORTEN/MINUTE) "
;G
2497 REM *****
2498 REM ** UEBERSETZEN **
2499 REM *****
2500 G=20/G*46
2510 FORI=1TOLEN(M$):A=ASC(MID$(M$,I,1))
2515 IFA=32THENGOTO2900
2520 FORJ=1TOLEN(M$(A)):B=ASC(MID$(M$(A),J,1))
2530 IFB=205THENGOSUB2700
2540 IFB=206THENGOSUB2800
2550 NEXTJ:FORK=1TOG:NEXTK
2560 NEXTI
2565 PRINT"BITTE TASTE DRUECKEN "
2570 POKE198,0:WAIT198,1:POKE198,0:GOTO300
2700 POKEW,33:FORK=1TOG:NEXT:POKEW,32:FORK=1TOG:NEXT:RETURN
2800 POKEW,33:FORK=1TO3*G:NEXT:POKEW,32:FORK=1TOG:NEXT:RETURN
2900 FORK=1TO3*G:NEXT:GOTO2560
2997 REM *****
2998 REM * MORSEZEICHEN EMPFANGEN *
2999 REM *****

```

```

3000 PRINT"GEBEN SIE JETZT IHRE MORSE- ZEICHEN UEBER DIE TASTE 'F7' EIN !"
3005 PRINT"WENN FERTIG, CA. 5 SEC WARTEN UND DANACH TASTE ERNEUT DRUECKEN"
3030 PRINT"ALLES ROGER ES KANN JETZT LOSGEHEN !"
3100 W=54276:I=49152
3110 POKE198,0:WAIT198,1:POKE198,0
3120 IFPEEK(203)<>64THEN3200
3160 POKEW,32:SYS65499
3170 IFPEEK(203)=64THEN3170
3180 GOTO3330
3200 POKEW,33:SYS65499
3210 IFPEEK(203)<>64THEN3210
3330 IFTI>255THEN3500
3340 POKEI,TI:I=I+1:GOTO3120
3497 REM *****
3498 REM * MORSE-AUSWERTUNG *
3499 REM *****
3500 PRINT"WES FOLGT DIE AUSWERTUNG":PRINT"X":M=0:M$=""
3510 FORII=49152TOI-1:M=M+PEEK(II):NEXT
3520 M=M/(I-49152)
3530 FORII=49152TOI-1:N=PEEK(II)
3540 IFN>MAND(II/2<>INT(II/2))THENGOSUB3600:M$=""
3550 IFN<MAND(II/2=INT(II/2))THENM$=M$+CHR$(205)
3560 IFN>MAND(II/2=INT(II/2))THENM$=M$+CHR$(206)
3570 NEXT:GOSUB3600
3580 PRINT:PRINT"BITTE TASTE DRUECKEN"
3590 POKE198,0:WAIT198,1:POKE198,0:GOTO3600
3600 FORJ=32TO90
3610 IFM$=M$(J)THENPRINTCHR$(J);
3620 NEXT
3630 IFN>M*3THENPRINTCHR$(32);
3640 RETURN
3997 REM *****
3998 REM * ZEICHENTABELLE *
3999 REM *****
4000 M=0:N=0:PRINT"X":FORI=1TO100:IFM$(I)=""THEN4100
4010 M$=M$(I):IFI=32THEN4100
4020 PRINTTAB(M*15)CHR$(I)" = "M$:N=N+1:IFN=19THENPRINT"S":N=0:M=M+1
4100 NEXT
4200 PRINT"X":FORK=1TO11:PRINTCHR$(17):NEXT:PRINT"BITTE TASTE DRUECKEN"
4300 POKE198,0:WAIT198,1:POKE198,0:GOTO300
4997 REM *****
4998 REM * ABGANG *
4999 REM *****
5000 PRINT"SPC(11)"
5010 FORI=1TO2000:NEXT:SYS64738
9998 REM *****
10000 POKE214,22:PRINT:POKE211,II/100
10010 PRINT" BITTE WARTEN "
10015 II=II+100
10020 RETURN
19997 REM *****
19998 REM * TITELBILD *

```



```

19999 REM *****
20000 POKE53280,0:POKE53281,0
20010 PRINT" "SPC(7)" " "SPC(20)" "
20020 PRINTSPC(8)" " "SPC(19)" "
20030 PRINTSPC(9)" " "SPC(19)" "
20040 PRINTSPC(8)" " "SPC(13)" "
20050 PRINTSPC(7)" " "SPC(12)" "
20060 PRINTSPC(12)" " "SPC(14)" "
20070 PRINTSPC(13)" " "SPC(13)" "
20080 PRINTSPC(14)" " "SPC(11)" "
20090 PRINTSPC(10)" " "SPC(11)" "
20100 PRINTSPC(27)" "
20110 PRINTSPC(11)" "MORSEZEICHEN"
20120 PRINTSPC(7)" "BY JENS MARTENS
UND":PRINTTAB(14)"STEFAN INGENHORST"
20130 PRINT"ANDROMAT - S
O F T W A R E"
20140 RETURN
29997 REM *****
29998 REM ** MORSE-ALPHABET **
29999 REM *****
30000 DATA32," "
30010 DATA34," ",39," "
30020 DATA40," ",41," "
30030 DATA43," ",45," "
30040 DATA46," ",47," "
30050 DATA48," ",49," "
30060 DATA50," ",51," "
30070 DATA52," ",53," "
30080 DATA54," ",55," "
30090 DATA56," ",57," "

```

```

30100 DATA44," ",61," "
30110 DATA63," ",65," "
30120 DATA66," ",67," "
30130 DATA68," ",69," "
30140 DATA70," ",71," "
30150 DATA72," ",58," "
30160 DATA73," ",74," "
30170 DATA75," ",76," "
30180 DATA77," ",78," "
30190 DATA79," ",80," "
30200 DATA81," ",82," "
30210 DATA83," ",84," "
30220 DATA85," ",86," "
30230 DATA87," ",88," "
30240 DATA89," ",90," ",-1
35000 REM *****
35010 REM SYS 65499 <=> TI$="000000"
35020 REM SYS 64738 <=> SYSTEM RESET
35030 REM *****
35040 REM VARIABLENTABELLE
35050 REM *****
35060 REM M$(X) <=> MORSEZEICHEN ZU X
35070 REM G <=> MORSEGESCHWINDIGKEIT
35080 REM M$,Z$ <=> HILFS-STRING
35090 REM W <=> SI:VOICE I: WAVEFORM
35100 REM A,B,I,II,K <=> HILFSZAEHLER
35110 REM TI <=> TIMER IN SEC/60
35120 REM *****
35130 REM ABSPEICHERUNG DER TIMERWERTE
35140 REM ZUM ENTSCHLUESSELN DER MORSE
35150 REM ZEICHEN AB 49152 (DEZ.)
35160 REM *****
35170 REM NEUER ZEICHENSATZ AB 12208
35180 REM *****
READY.

```

Listing vom »Morsetrainer« (Schluß)

9
 10
 40-130
 140-180
 190-220
 290-380
 990-1070
 1997-2120
 2497-2900
 2997-3340
 3497-3640
 3997-4300
 4997-5010
 10000-20140
 29997-30240
 35000-35180

Sprung zum Titelbild
 Begrenzung des Basicspeichers (sonst
 überlappen Zeichensatz und Variablen)
 Morsezeichen werden kreiert (77-kurz,
 78-lang)
 Soundchip wird initialisiert
 Variablenfeld wird mit Morsealphabet
 gefüllt
 Menü
 Zufallsbuchstabenblocks à 5 Zeichen
 werden kombiniert
 Eingebener Text wird in Morsealpha-
 bet übersetzt
 Morsezeichen werden hörbar gemacht
 Gemorste Zeichen werden ab 49152 ge-
 speichert
 Gespeicherte Timerwerte werden aus-
 gewertet
 Morsezeichentabelle auf Bildschirm
 Programmende mit Systemreset
 Titelbildunterprogramme
 Morsealphabet in Data-Zeilen
 Remarks zur Erklärung

Erläuterung des Programms nach Zeilennummern
 aufgeschlüsselt

MOVEMASTER

»Movemaster« wurde für die Erstellung beweglicher Grafiken konzipiert und läuft mit mindestens 8 KByte Speichererweiterung. Das Programm verwaltet bis zu 47 kleine Bilder, die man beliebig untereinander mischen, kopieren, speichern, laden und schließlich der Reihe nach, mit wählbarer Geschwindigkeit, im Bildschirm einblenden kann. Dies ergibt schließlich den Zeichentrickeffekt.

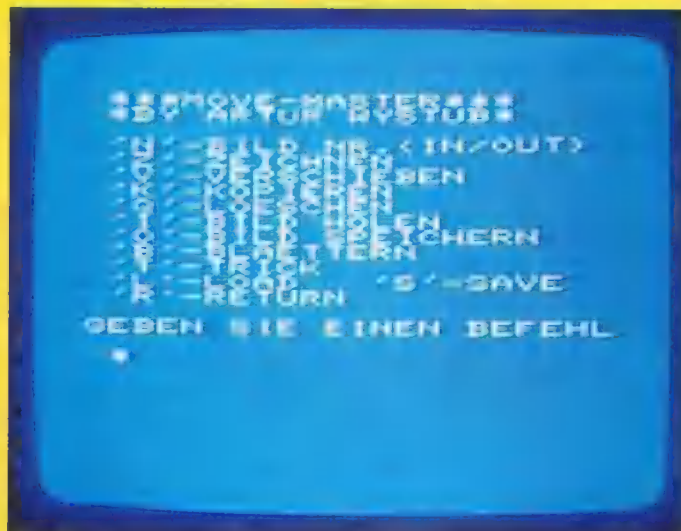


Bild 1. Das Movemaster-Hauptmenü

Zeichnungen können mit Joystick oder Tastatur in die Bilder eingebracht werden. Es ist auch möglich, die Lage der Bilder im Verhältnis zum Bildrahmen zu verändern, was die Entwicklung von Laufschrift oder ähnliche Anwendungen erheblich erleichtert. Das Programm »Movemaster« enthält Maschinencode-Routinen, die Aufgaben übernehmen, deren Ausführungszeit von Bedeutung ist (zum Beispiel Grafikausgabe). Außerdem nehmen diese Maschinencode-Routinen entscheidend weniger Speicherplatz ein als gleichwertige Basic-Routinen. Die SYS-Adressen der einzelnen Routinen gehen aus Tabelle 2 hervor.

Programmeingabe

Zuerst wird der Lader (Listing 1) eingetippt und sicherheitshalber abgespeichert. Eine kurze Zeit nach dem Starten des Laders erscheint auf dem Bildschirm: »FEHLERLOS« oder »TIPP-FEHLER«. Im zweiten Fall müssen sowohl die DATA-Zeilen als auch der Lader überprüft und verbessert werden. Erst bei der Anzeige »FEHLERLOS« darf der

Lader mit NEW gelöscht werden. Bei richtiger Eingabe beinhalten die Speicherzellen 45 und 46 die Zahlen 68 und 27 (dezimal). Ist das nicht der Fall, so gibt es noch einen (oder mehrere) Tippfehler, die bereinigt werden müssen. Das ganze Programm muß exakt übernommen werden — kein Zeichen mehr oder weniger.

Damit »Movemaster« zusammen mit den Maschinencode-Routinen abgespeichert werden kann, müssen davor die Speicherzellen 45 und 46 auf 0 und 30 geändert werden. Das hat den Vorteil, daß der Lader nicht mehr benötigt wird.

Programmbedienung

Nach dem Starten des Programms erscheint auf dem blauen Bildschirm das Hauptmenü. Es stehen jetzt folgende Befehle zur Verfügung:

- N** — Wählen der Bildnummer (von 1 bis 47 Bildern bei 24 KByte Speichererweiterung) oder Information über die momentane Bildnummer
- Q** — Löschen eines Bildes

oder des Arbeitsspeichers

- I** — Ein Bild in den Arbeitsspeicher übertragen und mit dem Inhalt mischen

- O** — Den Arbeitsspeicher in ein Bild übertragen und überschreiben

- K** — Ein Bild in ein anderes kopieren

- Z** — Bilder mit dem Joystick oder der Tastatur in den Arbeitsspeicher einzeichnen

- V** — Den Inhalt des Arbeitsspeichers mit dem Joystick oder der Tastatur verschieben

- B** — Den Inhalt des Arbeitsspeichers in das momentane Bild überschreiben und die momentane Bildnummer um eins erhöhen

- T** — Eine Bildfolge nacheinander abrufen und im Bildschirm einblenden

- L** — Eine Bildfolge vom Band lesen

- S** — Eine Bildfolge auf Band speichern

- R** — Rückkehr in das Hauptmenü


Zum Zeichnen wird, wie schon erwähnt, der Joystick oder die Tastatur eingesetzt: Punkt links — 1 bzw. Hebel links
Punkt hoch — 3 bzw. Hebel hoch
Punkt runter — 5 bzw. Hebel runter
Punkt rechts — 7 bzw. Hebel rechts
Punkt löschen — SHIFT bzw. Feuerknopf

Beispiele zur Programmbedienung

◆ Ein Rechteck soll ins Bild Nummer 1 eingezeichnet werden. Zuerst wird der Befehl N eingegeben, dem eine Frage »BILDNUMMER?« folgt, die mit 1 beantwortet wird. Mit der Eingabe von Z wird in den Grafikmodus umgeschaltet. Der Bildschirm wird schwarz. Die momentane Position ist in der Mitte des Bildschirms. Jetzt kann gezeichnet werden. Ist das Bild fertig, so wird es mit B abgespeichert. R führt zur Rückkehr zum Hauptmenü.

◆ Das Rechteck aus Bild Nummer 1 soll in Bild Nummer 2 etwas höher liegen.

Der VC 20 als Trickfilm studio



Hex		Dez
1000	Bildschirm-Speicher	4096
1200	Basicprogramm	4608
1B44	Variable	
1C1D	Maschinencode-Routinen	6980 Variablezeiger nach dem Start 7197
1E00	Arbeitsspeicher	7680 Variablezeiger vor dem Start
2000	Bild Nr. 1	8192 Bk. 1 Anfangsadresse
2200	Bild Nr. 2	8704
2400	Bild Nr. 3	9216
2600	Bild Nr. 4	9728
2800	Bild Nr. 5	10240
2A00	Bild Nr. 6	10752
2C00	Bild Nr. 7	11264
2E00	Bild Nr. 8	11776
3000	Bild Nr. 9	12288
3200	Bild Nr. 10	12800
3400	Bild Nr. 11	13312
3600	Bild Nr. 12	13824
3800	Bild Nr. 13	14336
3A00	Bild Nr. 14	14848
3C00	Bild Nr. 15	15360
3E00	Strings	15872
4000		16384 Bk. 2 Anfangsadresse

Speicherbelegung von »Movemaster«

Mit Z wird wieder in den Grafikmodus umgeschaltet. Im Arbeitsspeicher befindet sich noch das Rechteck, und es wäre unklug, es mit Q zu löschen, da es ja den Befehl V zur Verschiebung gibt. Nach der Eingabe von V blinken die Begrenzungspunkte, und das Rechteck kann verschoben werden. Vor dem Abspeichern mit B wird mit Z die Verschiebung unterbrochen. Die Eingabe von N führt zur Information über die momentane Bildnummer. Mit R wird die Grafik-Routine verlassen.

◆ Die Bilder Nummer 1 und Nummer 2 sollen miteinander gemischt und als Bild Nummer 3 abgespeichert werden. Im Arbeitsspeicher befindet sich noch das Rechteck Nummer 2 also reicht der Befehl I und I schon aus, um die Bilder zu mischen. Mit O und 3 wird der Arbeitsspeicher ins Bild Nummer 3 kopiert.

◆ Bild Nummer 2 soll in Bild Nummer 4 kopiert werden.

Dazu wird der Befehl K benutzt. Die erste Zahl, die eingegeben wird, ist logischerweise 2 gefolgt von einer 4.

◆ Die eben gezeichneten Bilder sollen der Reihe nach in den Bildschirm eingeblendet werden. Nach der Eingabe von T wird die Anzahl der Bilder angegeben, die proj-

ziert werden sollen, in diesem Falle also 4. Die Werte für die Verzögerung sollten zwischen 0 und 200 liegen, da sonst die Pausen zwischen den einzelnen Bildern zu lang werden und somit der Zeichentrickeffekt nicht erzielt wird. Mit R wird die Projektion unterbrochen.

(Artur Wystub)



Soiche Bilder
lassen sich mit
Movemaster per
Joystick erzeugen
und zum Leben
erwecken



HEX	DEZ	
IC1D	7197	Begrenzungspunkte löschen
IC3E	7230	Begrenzungspunkte setzen
IC4F	7247	Bild nach oben verschieben
IC81	7297	Bild nach unten verschieben
ICB1	7345	Bild nach rechts verschieben
ID08	7432	Bild nach links verschieben
ID5F	7519	Bild in Arbeitsspeicher kopieren
ID7E	7550	Arbeitsspeicher ins Bild kopieren
ID9D	7581	Grafikmodus setzten
IDB9	7609	Arbeitsspeicher löschen
IDD8	7640	Bild mit Arbeitsspeicher mischen

Adressen der Maschinencode-Routinen

AS	— Befehlsvariable
NA\$	— Bildfolgenname
BZ	— Höchstmögliche Bildnummer
BM	— Momentane Bildnummer
BO	— Gewählte Bilderzahl
GG	— Verzögerungsvariable
Z	— VIC-C-Basicadresse (36864)
X	— Horizontale Position auf dem Grafikbildschirm
Y	— Vertikale Position auf dem Grafikbildschirm
P	— Adresse der durch (X,Y) Position ermittelten Zelle
W	— Wert dieser Zelle
V	— Modusvariable (V=0 zeichnen, V=1 verschieben)
T	— Laufvariable in allen Schleifen

Variablenliste

```

10 REM***LOADER***
20 FORT=7197T07672
30 READ A$
40 H=ASC(MID$(A$,1,1))-48
50 L=ASC(MID$(A$,2,1))-48
60 IFH>9THENH=H-7
70 IFL>9THENL=L-7
80 POKET,H*16+L:Z=Z+H*16+L
90 NEXT:IFZ<>49265THENPRINT"TIPPFehler":END
95 PRINT"FEHLERLOS":END
100 DATA9,7F,2D,00,1E,8D,00,1E,A9,7F,2D,47,1E,8D,47,1E,A9,FE,2D
110 DATA80,1F,8D,B0,1F,A5,FE,2D,F7,1F,8D,F7,1F,60,A9,80,8D,00,1E
120 DATA8D,47,1E,A9,01,8D,B0,1F,8D,F7,1F,60,A2,01,BD,01,1E,9D,00
130 DATA1E,BD,49,1E,9D,48,1E,BD,91,1E,9D,90,1E,BD,D9,1E,9D,D8,1E
140 DATA8D,21,1F,9D,20,1F,BD,69,1F,9D,68,1F,BD,B1,1F,9D,B0,1F,E8
150 DATAE0,47,D0,D1,60,A2,45,BD,00,1E,9D,01,1E,BD,48,1E,9D,49,1E
160 DATA8D,90,1E,9D,91,1E,BD,D8,1E,9D,D9,1E,BD,20,1F,9D,21,1F,BD
170 DATA68,1F,9D,69,1F,BD,B0,1F,9D,B1,1F,CA,10,D3,60,A0,7F,A9,00
180 DATAA2,00,5E,B0,1F,5E,68,1F,90,07,98,7D,B0,1F,9D,B0,1F,5E,20
190 DATA1F,90,07,98,7D,68,1F,9D,68,1F,5E,D8,1E,90,07,98,7D,20,1F
200 DATA9D,20,1F,5E,90,1E,90,07,98,7D,D8,1E,9D,D8,1E,5E,48,1E,90
210 DATA07,98,7D,90,1E,9D,90,1E,5E,00,1E,90,07,98,7D,48,1E,9D,48
220 DATA1E,E8,E0,47,D0,B0,60,A0,00,A9,00,A2,00,1E,00,1E,1E,48,1E
230 DATA90,07,98,7D,00,1E,9D,00,1E,1E,90,1E,90,07,98,7D,48,1E,9D
240 DATA48,1E,1E,D8,1E,90,07,98,7D,90,1E,9D,90,1E,1E,20,1F,90,07
250 DATA98,7D,D8,1E,9D,D8,1E,1E,68,1F,90,07,98,7D,20,1F,9D,20,1F
260 DATA1E,B0,1F,90,07,98,7D,68,1F,9D,68,1F,E8,E0,47,D0,B0,60,A9
270 DATA00,85,03,A9,1E,85,04,A0,00,B1,01,91,03,E6,01,E6,03,D0,F6
280 DATAE6,02,E6,04,A9,20,C5,04,D0,EC,60,A9,00,85,03,A9,1E,85,04
290 DATAA0,00,B1,03,91,01,E6,01,E6,03,D0,F6,E6,02,E6,04,A9,20,C5
300 DATA04,D0,EC,60,A2,00,A9,40,A0,00,9D,00,10,18,69,09,E8,C8,C0
310 DATA07,D0,F4,A0,00,38,E9,3E,C9,49,D0,EB,60,A9,00,85,03,A9,1E
320 DATA85,04,A9,00,A0,00,91,03,E6,01,E6,03,D0,F8,E6,02,E6,04,A9
330 DATA20,C5,04,D0,EA,60,A9,00,85,03,A9,1E,85,04,A0,00,B1,01,11
340 DATA03,91,03,E6,01,E6,03,D0,F4,E6,02,E6,04,A9,20,C5,04,D0,EA
350 DATA60

```

READY.


```

10 REM*BY ARTUR WYSTUB
20 POKE45,68:POKE46,27:POKE47,68:POKE48,27:POKE49,68:POKE50,27
30 BZ=INT((PEEK(56)-34)/2):Z=36864:BM=1
40 GOSUB530:PRINT " 'N'-BILD NR.(IN/OUT)", " 'Z'-ZEICHNEN"
50 PRINT " 'Y'-VERSCHIEBEN", " 'K'-KOPIEREN", " 'Q'-LOESCHEN"
60 PRINT " 'I'-BILD HOLEN", " 'O'-BILD SPEICHERN", " 'B'-BLAETTERN"
70 PRINT " 'T'-TRICK ", " 'L'-LOAD", " 'S'-SAVE":GOSUB660
80 IFA$="N"THENGOSUB620:GOTO40
90 IFA$="Z"THENY=0:GOSUB210:GOTO40
100 IFA$="B"THENGOSUB210:GOTO40
110 IFA$="Y"THENY=1:GOSUB210:GOTO40
120 IFA$="K"THENGOSUB620:SYS7519:GOSUB620:SYS7550:GOTO40
130 IFA$="I"THENGOSUB620:SYS7519:GOSUB620:SYS7550:GOTO40
140 IFA$="O"THENGOSUB620:SYS7519:GOSUB620:SYS7550:GOTO40
150 IFA$="Q"THENGOSUB620:SYS7519:GOSUB620:SYS7550:GOTO40
160 IFA$="L"THENGOSUB460:GOTO40
170 IFA$="S"THENGOSUB410:GOTO40
180 IFA$="T"THENGOSUB350:GOTO40
190 IFA$="R"THEN40
200 GOSUB650:GOTO40
210 GOSUB560:X=27:Y=35
220 IFV=1THENSYS7197
230 GOSUB700:SYS7230:P=7751+INT(X/8)*72-Y:W=2+(7-(X-INT(X/8)*8))
240 POKEP,WORPEEK(P)
250 IF(PEEK(37137)AND32)=0ORPEEK(653)=1THENPOKEP,255-WANDPEEK(P)
260 IFV=1THENPOKEP,255-WANDPEEK(P)
270 GETA$:IFA$="N"THENGOSUB530:GOSUB600:GOSUB510:GOSUB590:GOTO210
280 IFA$="Q"THENGOSUB510:SYS7609:GOTO210
290 IFA$="Z"THENY=0:GOSUB510
300 IFA$="Y"THENY=1:GOSUB510
310 IFA$="B"THENGOSUB510:GOSUB640:SYS7197:SYS7550:BM=BM+1:SYS7230
320 IFBM>BZTHENGOSUB510:GOSUB640:SYS7197:SYS7550:BM=BM+1:SYS7230
330 IFA$="R"THENSYS7197:GOSUB510:RETURN
340 GOTO220
350 INPUT"NBILDANZAHL:1-";BO:GOSUB510
360 IFBO<1ORBO>BZTHENGOSUB800:RETURN
370 INPUT"VERZOEGERUNG:";GG:GG=ABS(GG):GOSUB510:GOSUB560
380 FORBM=1TOBO:FORT=0TOGG:NEXTT:GOSUB640:SYS7519:NEXTBM
390 GETA$:IFA$="R"THENGOSUB510:RETURN
400 GOTO380
410 INPUT"NBILDANZAHL";BO:GOSUB510
420 IFBO<1ORBO>BZTHENGOSUB800:RETURN
430 INPUT"NAME";NA$:GOSUB510:PRINT" "
440 OPEN1,1,1,NA$:PRINT#1,NA$:PRINT#1,BO:FORBM=1TOBO:FORT=0TO511
450 PRINT#1,PEEK((BM*2+30)*256+T):NEXTT,BM:CLOSE1:GOSUB510:RETURN
460 INPUT"NAME ";NA$:GOSUB510
470 OPEN1,1,0,NA$:INPUT#1,BO
480 PRINT"GEFUNDEN ";NA$:PRINT"NBILDANZAHL=";BO
490 FORBM=1TOBO:FORT=0TO511:INPUT#1,W:POKE(BM*2+30)*256+T,W
500 NEXTT,BM:CLOSE1
510 POKE36878,15:FORT=0TO10:POKE36876,254:POKE36876,244:NEXT
520 POKE36878,0:RETURN
530 POKEZ,12:POKEZ+1,38:POKEZ+2,22:POKEZ+3,46:POKEZ+5,192
540 POKEZ+15,110
550 PRINT"***MOVE-MASTER***:PRINT" *BY ARTUR WYSTUB*":RETURN
560 PRINT" "
570 POKEZ,25:POKEZ+1,61:POKEZ+2,7:POKEZ+3,18:POKEZ+5,207
580 POKEZ+15,8:SYS7581:RETURN
590 FORT=0TO2000:NEXT:RETURN
600 PRINT"NBILD NUMMER:";BM
610 PRINT"MAX. BILDANZAHL:";BZ:RETURN
620 INPUT"NBILD NUMMER:";BM:GOSUB510
630 IFBM<1ORBM>BZTHENGOSUB800:BM=1:GOTO40
640 POKE1,0:POKE2,BM*2+30:RETURN
650 PRINT"BEFEHL UNBEKANNT":GOSUB510:GOSUB590:RETURN
660 PRINT" 'R'-RETURN", "GEBEN SIE EINEN BEFEHL":PRINT" * ";
670 GETA$:IFA$=""THEN670
680 PRINTA$:GOSUB510
690 RETURN
700 POKE37154,127:POKE37139,0
710 IF(PEEK(37137)AND16)=0ORPEEK(203)=0THENX=X-1:IFV=1THENSYS7432
720 IF(PEEK(37137)AND128)=0ORPEEK(203)=3THENX=X+1:IFV=1THENSYS7345
730 IF(PEEK(37137)AND4)=0ORPEEK(203)=1THENY=Y+1:IFV=1THENSYS7247
740 IFX<0THENX=0
750 IFX>55THENX=55
760 IFX>55THENY=0
770 IFY<1THENY=1
780 IFY>70THENY=70
790 POKE37154,255:RETURN
800 PRINT"UNMOEGLICH":GOSUB510:GOSUB590:RETURN

```

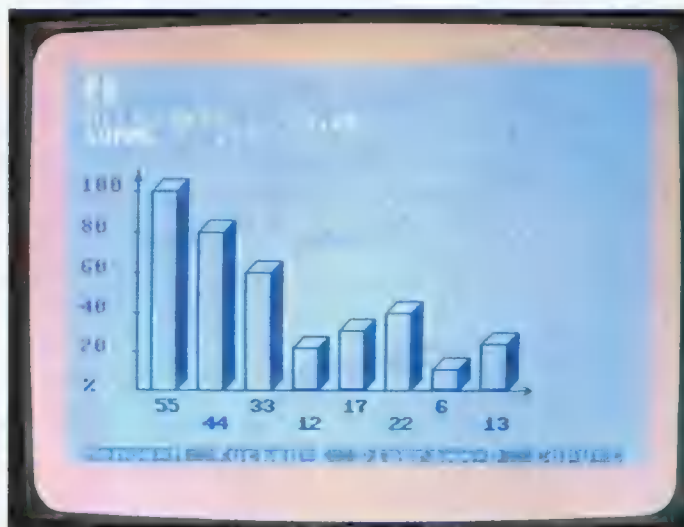



Bild 1. So übersichtlich erscheint die Darstellung eines Balkendiagramms auf dem Bildschirm

**Hardcopy des
Bildschirminhalts
auf einem Epson
RX-80-Drucker
erstellen lassen.**

Nach dem Start meldet sich das Programm mit dem Titel »Chart 1 — Programm zur grafischen Zahlendarstellung«. Dieser Titel erscheint etwa drei Sekunden. Durch Drücken der Return-Taste wird vorzeitig zum Hauptmenü gesprungen. Bei der Hauptmenüabfrage akzeptiert das Programm nur Zahlen von 1 bis 4 und den Buchstaben E.

Vom Hauptmenü aus sind alle Funk- tionen abrufbar

Zunächst zur Funktion 4. Durch diese Funktion wird die letzte im Speicher befindliche Grafik dargestellt. Falls sich auch noch die Werte im Hauptspeicher befinden, kann von dieser Grafik ausgehend das Programm durch Antippen des entsprechenden Buchstabens fortgesetzt werden. Falls die Absolutsumme (MC) Null ist, geht das Programm davon aus, daß keine Werte mehr

Grafische Darstellung

statt Zahlenfriedhof

Übersichtliche Grafik statt endloser Zahlenkolonnen — dieses Programm für den Commodore 64 mit Simons Basic kann Tabellen wahlweise als Balken-, Kurven- oder konventionelle Strichdiagramme darstellen. Die laufende Umschaltung zwischen den verschiedenen Darstellungsformen ist möglich. Auf Tastendruck kann man sich jederzeit eine

```

35 REM*****
40 REM**** CHART 1 *****
70 REM*****
75 POKE53280,10:POKE53281,12:PRINT"1"
80 DIMX(22),A(22),B(22),C(22),F(22)
100 PRINT"J"AT(12,8)">>> CHART 1 <<<"
130 PRINTAT(7,11)"PROGRAMM ZUR GRAPHISCHEN"
160 PRINTAT(11,13)"ZAHLENDARSTELLUNG"
170 PAUSE3
220 PRINT"JA DARSTELLUNGSFORMEN"
250 PRINTAT(5,5)"KUCHENGRAPHIK.....1"
280 PRINTAT(5,7)"BALKENGRAPHIK.....2"
310 PRINTAT(5,9)"DIAGRAMM.....3"
320 PRINTAT(5,11)"LETZTE GRAPHIK.....4"
340 PRINTAT(5,15)"BITTE WAEHLEN SIE >>>"
350 PRINTAT(5,19)"ENDE.....E"
370 GETE$:IFE$=""THEN370
375 IFE$="E"THENPRINT"J":END
377 IFE$="4"THENGOTO 3100
380 IFE$<"1"ORE$>"3"THEN370
400 PRINTAT(28,15)E$
430 E=VAL(E$)
440 PAUSE1
450 GOSUB5000
460 ONEGOTO1000,2000,4000
1000 REM*****
1010 REM***** KUCHEN *****
1020 REM*****
1100 I=360/MC:MAX=-10138
1110 HIRES0,14
1111 LOW COL1,14,14
1112 CIRCLE160,110,80,40,1
1114 ARC160,130,90,270,5,80,40,1
1116 LINE80,110,80,130,1
1118 LINE240,110,240,130,1
1120 HI COL:S=0:W1=0
1122 FORD=1TOA:F(D)=INT(ABS(X(D))+.5):NEXTD
1130 FORD=1TOA
1140 S=S+F(D)
1150 W1=((S*I)/180)*PI
1152 W3=((F(D)/2)*I)/180*PI
1155 LOW COL1,14,14
1160 ANGL160,110,S*I,80,40,1
1161 IF(S*I)<270AND(S*I)>90THENGOSUB1900
1163 HI COL
1164 J=115:K=67
1165 IFA>10THENJ=120
1166 IFX(D)<0THENLOW COL 2,14,14
1167 IFX(D)=0THEN1180
1168 IFMC/ABS(X(D))>75THEN1180
1169 D$=LEFT$(STR$(X(D)),3)
1170 TEXT155-COS((W1-W3)+(PI/2))*J,110-SIN((W1-W3)+(PI/2))*K,D$,1,1,8
1180 IFX(D)<0THENHI COL
1181 NEXTD
1183 W1=0:W2=0:W3=0:S=0
1184 FORD=1TOA
1185 IFX(D)>MAXTHENMAX=X(D)
1186 NEXTD
1187 W1=0:W2=0:W3=0:S=0

```

Listing für die grafische Darstellung von Zahlen


```

1188 FORD=1TOR
1189 S=S+F(D)
1190 LOW COL 1,14,14
1191 W2=((S*I)/180)*PI:W3=((F(D)/2)*I)/180)*PI
1193 IFX(D)=MAXANDX(D)=X(D+1)THEN1199
1195 IFX(D)=MAXTHEN PRINT160-COS(W2-W3)+PI/2)*20,110-SIN(W2-W3)+PI/2)*20,1
1198 NEXTD
1199 HI COL
1200 GOSUB1925
1207 LOW COL11,14,14
1210 TEXT10,6,"Mittelwert=";T$;1,2,12
1215 IFLEFT$(M$,1)<>"J"THEN1400
1220 TEXT10,25,"SUMME=";STR$(M$),1,1,8
1230 TEXT10,33,"SUMME=";STR$(M$),1,1,8
1240 HI COL
1400 H$="MENUE\ IALKEN=I -IAGRAM=- -OPY=-"
1450 LOW COL 13,14,14
1500 TEXT3,190,"M"+H$,1,1,8
1550 HI COL
1700 GETZ$:IFZ$=""THEN1700
1730 IFZ$<"M"ANDZ$<"B"ANDZ$<"D"ANDZ$<"C"THEN1700
1760 IFZ$="M"THENCSET0:GOTO220
1790 IFZ$="D"THENGOTO4000
1820 IFZ$="B"THENGOTO2000
1850 IFZ$="C"THENCOPY:GOTO1700
1880 END
1900 D1=160-COS(W1+PI/2)*80:D2=110-SIN(W1+PI/2)*40:D3=D2+20
1910 LINED1,D2,D1,D3,1
1920 RETURN
1925 I=360/MC:S=0:LOW COL1,14,14
1930 FORD=1TOR
1931 S=S+F(D)
1932 IFX(D)=MAXANDX(D)=X(D+1)THENRETURN
1933 W2=((S*I)/180)*PI:W3=((F(D)/2)*I)/180)*PI:W4=((S-F(D))*I)/180)*PI
1935 IF$1<90AND(S-F(D))*I<90ANDX(D)=MAXTHEN1968
1938 IF$1<265AND(S-F(D))*I<265ANDX(D)=MAXTHEN1968
1941 IF$1<270AND(S-F(D))*I<90ANDX(D)=MAXTHEN1942:ELSE:GOTO1944
1942 PRINT160-COS(W2-W3)+PI/2)*80,115-SIN(W2-W3)+PI/2)*40,1:GOTO1968
1944 IF$1<93AND(S-F(D))*I<90ANDX(D)=MAXTHEN1945:ELSE:GOTO1947
1945 PRINT161-COS(W2+PI/2)*80,115-SIN(W2+PI/2)*40,1:GOTO1968
1947 IF$1<270AND(S-F(D))*I<265ANDX(D)=MAXTHEN1948:ELSE:GOTO1968
1948 PRINT159-COS(W4+PI/2)*80,115-SIN(W4+PI/2)*40,1:GOTO1968
1950 IF$1<270AND(S-F(D))*I<90ANDX(D)=MAXTHEN1952:ELSE:GOTO1968
1952 PRINT160-COS(W2-W3)+PI/2)*80,115-SIN(W2-W3)+PI/2)*40,1:GOTO1968
1968 NEXTD
1970 HI COL
1980 RETURN
1990 END
2000 REM*****
2010 REM***** BALKENGRAPHIK *****
2040 REM*****
2050 Y=0:Z=0:C=0
2070 FORD=1TOR
2100 A(D)=X(D)
2130 NEXT
2230 FORD=1TOR
2240 IFABS(A(D))>YTHENY=ABS(A(D))
2250 NEXTD
2252 GH=ABS(Y)/100
2255 IFMC<MBANDAC=10THENC=50:GH=2*GH

```

Listing für die grafische Darstellung von Zahlen (Fortsetzung)

vorhanden sind und springt in das Hauptmenü zurück.

Die Werteingabe ist unkompliziert

Bei der Auswahl von Zahlen zwischen 1 und 3 wird zur Werteingabe übergegangen. Als erstes wird der Titel abgefragt, wobei ein Leerstring nicht akzeptiert wird, da der Titel gegebenenfalls gleichzeitig als Filename verwendet wird. Bei der darauffolgenden Abfrage nach der Mittelwert- und Summenberechnung werden nur Eingaben akzeptiert, de-

ren jeweils am meisten links stehendes Zeichen ein j oder n ist.

Anschließend wird die Anzahl der Werte abgefragt, die zwischen 2 und 20 liegen muß. Bei der jetzt folgenden Werteingabe sind alle Zahlen zulässig, die im Wertebereich des Commodore 64 liegen. Nach der Darstellung der Werte in zwei Reihen muß die Sicherheitsabfrage in der gegebenen Weise mit j oder n beantwortet werden. Das Programm bietet jetzt die Möglichkeit, die Wertetabelle auszudrucken und/oder abzuspeichern.

Achtung: Bei der Abspeicherung wird ein eventuell

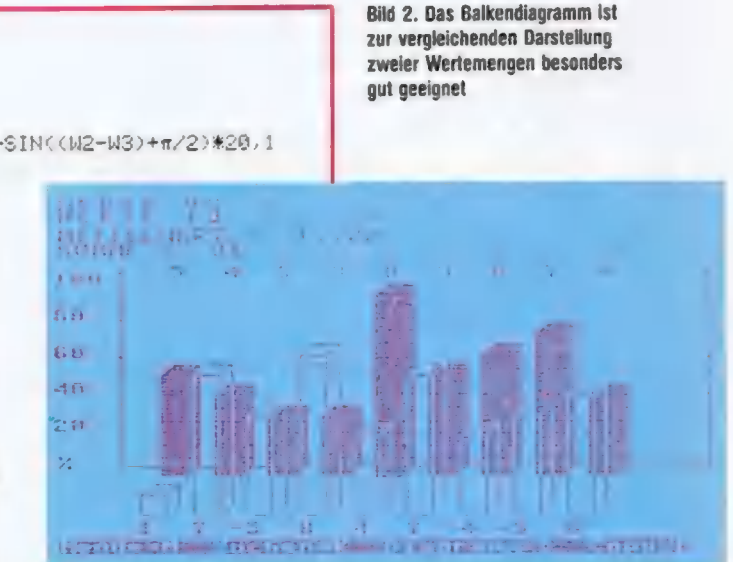


Bild 2. Das Balkendiagramm ist zur vergleichenden Darstellung zweier Wertemengen besonders gut geeignet

Anschließend wird in den Grafikmodus umgeschaltet.

Die Werte werden — ausgehend von der »12-Uhr-Stellung« — als »Kuchenstücke« dargestellt. Der Maximalwert (oder die Maximalwerte, falls sie nicht direkt nebeneinanderliegen) werden ausgefüllt. Die Werte werden konzentrisch um den Kuchen ausgegeben, wobei negative Werte rot dargestellt werden und Null-Werte, ebenso wie Werte, die klei-

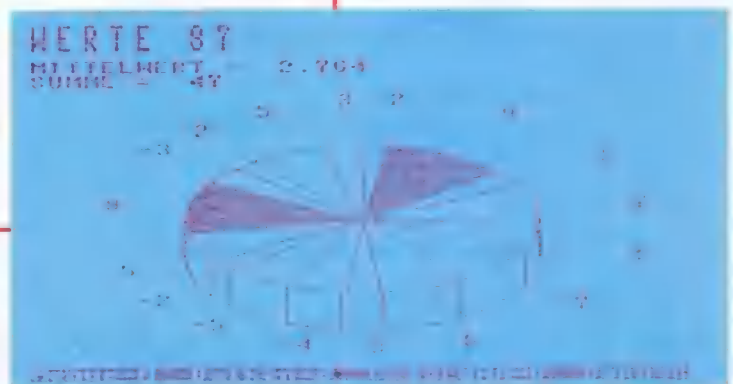


Bild 3. Mit dem Kuchendiagramm lassen sich Größenverhältnisse innerhalb einer Wertemenge sehr übersichtlich und augenfällig darstellen

unter diesem Namen existierendes File überschreiben. Die so abgespeicherte Tabelle kann später wieder mit dem SCRLD-Befehl des Simons Basic aufgerufen werden. Nach der Abspeicherung der Tabelle wird der Floppyfehlerkanal abgefragt und gegebenenfalls eine Fehlermeldung ausgegeben.

ner als $\frac{1}{2}$ der Betragssumme MC sind, nicht berücksichtigt werden.

Balkendiagramme

Bis zu zehn Werte werden einreihig dargestellt. Bei negativen Werten werden die Balken sowohl über als auch


```

2256 FORD=1TOR:A(D)=A(D)/GH:NEXTD
2260 IFA<=10THEN2300
2290 IFA>10THEN2800
2300 REM***** MAX 10 WERTE *****
2310 HIRES0,14
2312 LOW COL 15,14,14
2315 TEXT10,5,"M"+T$,1,2,10
2317 IFLEFT$(M$,1)="J"THENTEXT10,23,"MITTELWERT = "+STR$(MW),1,1,8
2318 IFLEFT$(M$,1)="J"THENTEXT10,31,"SUMME = "+STR$(MB),1,1,8
2319 HI COL
2320 LINE40,160-C,(40+A*26)+10,160-C,1
2322 LINE(40+A*26)+5,158-C,(40+A*26)+10,160-C,1
2324 LINE(40+A*26)+5,162-C,(40+A*26)+10,160-C,1
2340 LINE40,160,40,50,1
2342 LINE42,55,40,50,1
2344 LINE38,55,40,50,1
2350 Q=0
2360 FORF=140-(C*(4/5))TO60STEP((-20)+(2*C))
2380 LINE38,F,41,F,1
2390 Q=Q+20
2395 LOW COL 9,14,14
2400 TEXT 0,F-6,STR$(Q),1,1,8
2402 HI COL
2405 NEXTF
2410 CHAR10,155,37,1,1
2430 FORD=1TOR
2432 H1=40+D*26
2435 IFX(D)<0THENLOW COL2,14,14
2436 IFA(D)<0THENV=110:ELSE V=110-A(D)
2437 IFMC=MBTHENV=160-A(D)
2440 RECH1=18,V,14,ABS(A(D)),1
2560 LINEH1=18,V,H1-12,V-6,1
2570 LINEH1=4,160-A(D)-C,H1+2,154-A(D)-C,1
2580 LINEH1=4,160-C,H1+2,154-C,1
2590 LINEH1+2,V-6,H1-12,V-6,1
2600 LINEH1+2,154-A(D)-C,H1+2,154-C,1
2605 IFX(D)=0THEN2615
2607 IFX(D)<0THENPRINTH1,108,1
2610 PRINTH1=3,V+1,1
2615 IFX(D)<0THENHI COL
2620 NEXT D
2625 T=-1
2626 LOOP
2628 T=T+2
2630 B(T)=INT(X(T)+.5)
2631 B(T+1)=INT(X(T+1)+.5)
2632 IFX(T)=0THEN2635
2634 P=105-A(T)-(A(T))/ABS(A(T))*10
2635 IFX(T+1)=0THEN2637
2636 Q=105-A(T+1)-(A(T+1))/ABS(A(T+1))*10
2637 IFMB=MCTHENP=165:Q=174
2638 IFA(T)<0THENLOW COL 2,14,14
2639 IFX(T)=0THEN2642
2640 TEXT(40+T*26)-23,P,LEFT$(STR$(B(T)),3),1,1,7
2642 EXIT IFT=A
2645 IFA(T+1)<0THENLOW COL 2,14,14
2648 IFX(T+1)=0THEN2655
2650 TEXT(40+(T+1)*26)-23,Q,LEFT$(STR$(B(T+1)),3),1,1,7
2655 EXIT IF T+1=A
2660 IFA(T)<0THENHI COL
2662 IFA(T+1)<0THENHI COL
2670 END LOOP
2690 GOTO3800
2800 REM***** MAX 20 WERTE *****
2820 HIRES0,14
2825 LOW COL 1,14,14
2830 TEXT10,03,"M"+T$,1,2,10
2840 IFLEFT$(M$,1)="J"THENTEXT10,20,"MITTELWERT = "+STR$(MW),1,1,8
2845 IFLEFT$(M$,1)="J"THENTEXT10,28,"SUMME = "+STR$(MB),1,1,8
2847 HI COL
2850 LINE40,150,40,40,1
2860 LINE315,150,315,40,1
2870 LINE40,150,315,150,1
2880 LINE38,45,40,40,1
2890 LINE42,45,40,40,1
2900 LINE313,45,315,40,1
2910 LINE317,45,315,40,1
2915 Q=0
2920 FORF=130TO50STEP-20
2930 LINE38,F,41,F,1
2940 LINE314,F,317,F,1
2945 Q=Q+20
2947 LOW COL 7,14,14
2950 TEXT0,F-6,STR$(Q),1,1,8
2955 CHAR10,145,37,1,1
2960 HI COL
2965 NEXT
2970 FORD=1TOR:A(D)=ABS(A(D)):NEXTD
3000 IFFRAC(A/2)<0THENB=(INT(A/2))+1:ELSE B=(A/2)
3010 FORD=B+1TOR
3015 H2=50+(D-B)*25

```

Listing für die grafische Darstellung von Zahlen (Fortsetzung)

Zeile	Funktion
79	Farbwahl für Textdarstellung
80	Dimensionierung der Feldvariablen
100-140	Hauptmenu
1000-1050	Kochen
1100-1115	Übersicht
1120-1160	Zeichnen des Kochengerüsts
1160	Verzweigung nach 1900
1164-1181	Einschreiben der Werte
1184-1190	Bestimmung des Maximalwertes
1187-1199	Ausfüllen des größten Sektors
1200	Verzweigung nach 1935
1205-1500	Titel, Mittelwert, Summe und Menu
1700-1850	Menüabfrage
1900-1930	Zeichnen der Begrenzung am Rande des Kochens
1935-1980	Ausfüllen der Maximumvorderfläche
2000-2050	Balkengrafik
2050-2090	Umrechnung in darstellbare Werte, Überprüfung auf Negativwerte
2600-2650	Darstellung von höchstens 10 Werten
2910-2916	Titel, Mittelwert, Summe
2920-2940	Koordinatensystem, Prozentzahlen
2950-2980	Zeichnen der Balkengerüste und Ausfüllen der rechten Seitenfläche
2980-2990	Einschreiben der Werte
2990-3000	Spring nach 1900
3000-3040	Darstellung von bis zu 20 Werten
3040-3047	Titel, Mittelwert, Summe
3050-3060	Koordinatensystem mit Prozentangabe
3070-3140	Zeichnen der hinteren Balken, Ausfüllen der Flächen und Einschreiben der Werte
3140-3142	Zeichnen der hinteren Balken, Einschreiben der Werte
3150-3160	Menüabfrage
4000-4060	Diagramm
4050-4060	Bestimmung des Maximalwertes
4210	Konstantenberechnung für darstellbare Größen
4270-4340	Titel, Mittelwert, Summe
4357	Umrechnung der Werte in Koordinaten
4360-4365	Koordinatensystem mit Prozentangaben
4365-4770	Diagrammdarstellung
4780-4880	Einschreibung der Werte (zweiseitig)
4900-4960	Menüabfrage
5000-5020	Werteingabe
5060-5070	Titel
5120-5161	Mittelwert-Summe-Abfrage
5160-5190	Anzahl der Werte (0-20)
5190-5420	Werteingabe
5500-5560	Wertetabelle
5640-5660	Sicherheitsabfrage
5700	Verzweigung nach 5900
5720-5750	Ausgabe von Mittelwert und Summe
5800-5900	Ausgabe der Wertetabelle auf dem Drucker
5910-6000	Speicherung der Tabelle
6000-6300	Mittelwert und Summe
6120	Berechnung der Summe der Beträge der Werte
6160	Berechnung der Summe der Werte
6180	Mittelwert
6190-6197	Formatierung von Mittelwert und Summe
9000-9060	Fahrtenplan der Floppy
9060-9100	Darstellung der leeren Grafik
9100	Darstellung
9140	Abfrage der Betragesumme
9140-9180	Menüabfrage

Zeilenorientierter Ablaufplan zu »Chart 1«

unter der X-Achse angezeigt (negative Werte wieder rot), ansonsten nur oberhalb der X-Achse. Nullwerte werden ohne Bezeichnung als leeres zweidimensionales Rechteck gezeichnet.

Bis zu 20 Werte werden zweireihig hintereinander dargestellt und zwar nur oberhalb der X-Achse. Das Vorzeichen der Werte (positiv oder negativ) wird nur durch die eingeschriebenen Werte kenntlich gemacht, wobei negative Werte wiederum rot erscheinen. Die Reihenfolge der Werte läuft von links vorne nach rechts hinten.


```

3020 RECH2=16,156-A(D),12,A(D),1
3030 LINEH2=15,150,H2-5,150,0
3040 LINEH2=16,156-A(D),H2-10,150-A(D),1
3050 LINEH2=4,156-A(D),H2+2,150-A(D),1
3060 LINEH2=4,156,H2+2,150,1
3070 LINEH2+2,150,H2+2,150-A(D),1
3080 LINEH2=10,150-A(D),H2+2,150-A(D),1
3090 LINEH2=3,150,H2+1,150,0
3092 IFA(D)=0THEN3340
3095 IF(Y/A(D))>75THEN3340
3097 IFX(D)=0THEN3340
3100 PRINTH2=15,157-A(D),1
3110 PRINTH2=9,155-A(D),1
3120 PRINTH2=3,154,1
3130 IFX(D)<0THENLOW COL 2,14,14
3140 TEXTH2=20,40,LEFT$(STR$(X(D)),3),1,1,8
3330 IFX(D)<0THENHI COL
3340 NEXT
3410 FORD=1TOB
3415 H3=50+D*25
3420 RECH3=26,178-A(D),12,A(D),2
3440 LINEH3=26,178-A(D),H3-20,172-A(D),2
3450 LINEH3=14,178-A(D),H3-8,172-A(D),2
3460 LINEH3=14,178,H3-8,172,2
3470 LINEH3=8,172,H3-8,172-A(D),2
3480 LINEH3=20,172-A(D),H3-8,172-A(D),2
3500 IFX(D)<0THENLOW COL 2,14,14
3510 TEXTH3=35,181,LEFT$(STR$(X(D)),3),1,1,8
3520 IFX(D)<0THENHI COL
3540 NEXT
3800 H$=" MENUE=\ KUCHEN=" DIAGRAMM=" COPY="
3810 LOW COL11,14,14
3820 TEXT3,191,"0"+H$,1,1,8
3830 HI COL
3900 GETZ: IFZ$="" THEN3900
3930 IFZ$<"M"ANDZ$<"K"ANDZ$<"D"ANDZ$<"C" THEN3900
3940 IFZ$="M" THENGOTO220
3950 IFZ$="D" THENGOTO4000
3960 IFZ$="K" THENGOTO1000
3970 IFZ$="C" THENCOPY:GOTO3900
3980 END
4000 REM *****
4030 REM ***** DIAGRAMM *****
4060 REM *****
4090 Y=0
4120 FORD=1TOA
4150 IFABS(X(D))>Y THENY=ABS(X(D))
4180 NEXT
4210 GH=Y/90
4270 HIRES1,14
4280 LOW COL11,14,14
4300 TEXT10,10,"I"+T$,1,2,10
4330 IFLEFT$(M$,1)="J" THENTEXT10,30,"MITTELWERT = "+STR$(MW),1,1,8
4340 IFLEFT$(M$,1)="J" THENTEXT10,38,"SUMME = "+STR$(MB),1,1,8
4345 HI COL
4350 IFMB<0 THENC=55:GH=GH*2
4357 FORD=1TOA:C(D)=X(D)/GH:NEXTD
4360 LINE50,150,50,50,1
4365 FORI=142TO54STEP-8:LINE49,I,51,I,1:NEXTI
4380 LINE48,55,50,50,1:LINE52,55,50,50,1
4390 FORI=60TO240STEP 10:LINEI,149-C,I,151-C,1:NEXTI
4420 LINE50,150-C,250,150-C,1
4440 LINE245,148-C,250,150-C,1:LINE245,152-C,250,150-C,1
4450 FORI=20TO100STEP20
4452 LOW COL 9,14,14
4455 IFMB<0 THEN4470
4459 Z=0
4460 FORI=135TO55STEP-20:Z=Z+20:TEXT10,I,STR$(Z),1,1,8:NEXTI:CHAR20,150,37,1,1
4465 GOTO4640
4470 Z=-120
4480 FORI=140TO45STEP-9:Z=Z+20:TEXT10,I,STR$(Z),1,1,8:NEXTI:CHAR20,155,37,1,1
4640 HI COL
4650 FORD=1TOA-1
4660 LINE(50+D*10)-10,150-C(D)-C,(50+D*10),150-C(D+1)-C,1
4770 NEXT
4780 D=-1
4790 IFMC<0 THENLL=2:ELSE:LL=3
4800 LOOP
4805 LOW COL3,14,14
4810 D=D+2
4815 IFX(D)<0THENLOW COL2,14,14
4820 TEXT27+D*10,160,LEFT$(STR$(X(D)),LL),1,1,8
4822 IFX(D)<0THENHI COL
4824 EXIT IF D=A
4826 LOW COL3,14,14
4828 IFX(D+1)<0THENLOW COL 2,14,14
4830 TEXT27+(D+1)*10,172,LEFT$(STR$(X(D+1)),LL),1,1,8
4832 IFX(D+1)<0THENHI COL
4834 EXIT IF D+1=A
4840 HI COL
4850 END LOOP

```

Listing für die grafische Darstellung von Zahlen (Fortsetzung)

X(22),A(22),B(22)
C(22),F(22):
E\$,E:
T\$:
M\$:
A\$,A:
D:
W\$:
MB:
MC:
MW:
MAX,YX:
V\$:
SS:
AA,BB,CC,DD:
Z\$:
GH:
LL:
I:
Z:
H\$:
S:
W1,W2,W3,W4:
D1,D2,D3:
J,K:
DS:
C:
H1,H2,H3:
F:
V:
P,Q:

Feldvariable für Zahlenwerte

Abfrage des Darstellungsmodus
Titel der Tabelle, SCRSV-Name
Abfrage nach Mittelwert-Summen-
Berechnung
Anzahl der Werte
Schleifenvariable
Sicherheitsabfrage, ob Werte in Ord-
nung sind
Summe der Werte
Summe der Absolutwerte
Mittelwert
Maxima der Feldvariablen
Abfrage nach dem Ausdruck der Werte-
tabelle
Abfrage nach der Speicherung der Wer-
tetabelle
Wird unter T\$ abgespeichert
Floppy-Fehlerkanal
Abfrage nach der Weiterführung des
Programms
Faktor zur Umrechnung der Feldvaria-
blen in darstellbare Größe
Länge des Textstrings beim Diagramm
Schleifenvariable
Darstellung der Prozentwerte beim Dia-
gramm
Textstring; unterste Zeile bei der Grafik
Winkelberechnung beim Kuchen
Winkel bei Kuchen (Bogenmaß)
Winkel beim Kuchen (Bogenmaß)
Koordinaten zur Berechnung der Zahlen
beim Kuchen
LEFT\$ vom Textstring (Kuchen)
Konstante für die Negativdarstellung
Laufvariablen beim Balken
Laufvariablen beim Balken
Koordinaten (pos.-neg.) beim Balken
Koordinaten beim Balken

Variablenliste von »Chart 1«

Strichdiagramm

Nach dem Zeichnen des Koordinatensystems — auch hier wird bei negativen Werten die X-Achse nach oben verschoben — wird der Diagrammzug eingezogen. Die Werte werden unterhalb der X-Achse in der üblichen Weise zweireihig eingeschrieben.

Kommandozeile

Durch Betätigen der jeweiligen Taste können die Werte entweder als Kuchen, Balken oder Strichdiagramm dargestellt werden. Ferner kann mit der Taste M zum Hauptmenü zurückgesprungen werden. Außerdem bietet sich noch die Möglichkeit, den Grafikbildschirm ausdrucken zu lassen. Diese Hardcopy-Routine ist auf den Epson RX-80-Drucker abgestimmt und muß bei Verwendung eines anderen


```

4900 H$=" MENUE=\ 'UCHEN=' IALKEN=I -OPY=-"
4902 LOW COL 6,14,14
4905 TEXT3,188,"I"+H$,1,1,8
4907 HI COL
4910 GETZ$:IFZ$=""THEN4910
4930 IFZ$<"M"ANDZ$<"K"ANDZ$<"B"ANDZ$<"C"THEN4910
4940 IFZ$="M"THENCSET0:GOTO220
4950 IFZ$="B"THENGOTO2000
4960 IFZ$="K"THENGOTO1000
4970 IFZ$="C"THENCOPY:GOTO4910
4980 END
5000 REM*****
5010 REM***** WERTEEINGABE *****
5020 REM*****
5030 PRINT"J" WERTEEINGABE
5060 PRINTAT(5,5)"TITEL DER TABELLE"
5090 INPUT"J"
5100 IFT$=""THEN5030
5120 PRINTAT(5,8)"MITTELWERT UND SUMME"
5150 INPUT"J"
5160 IFLEFT$(M$,1)<"J"ANDLEFT$(M$,1)<"N"THEN5120
5180 PRINTAT(5,11)"ANZAHL DER WERTE"
5210 INPUT"J"
5240 A=VAL(A$)
5270 IFAC20RA>20THENPRINTAT(6,13)"MAXIMAL 20 WERTE":GOTO5180
5330 FORD=1TOA:X(D)=100*(X(D)):X(D)=INT(X(D)):X(D)=X(D)/100
5360 PRINTAT(5,17)"WERT";D;" ="
5390 INPUT"J"
5420 NEXTD
5500 PRINT"J" WERTETABELLE
5530 PRINTAT(5,3)T$
5550 PRINT"J"
5560 FORD=1TOASTEP2:X(D)=100*(X(D)):X(D)=INT(X(D)):X(D)=X(D)/100
5570 X(D+1)=100*(X(D+1)):X(D+1)=INT(X(D+1)):X(D+1)=X(D+1)/100
5590 PRINT" WERT"(D)=""LEFT$(STR$(X(D)),7)
5600 PRINT" J WERT"(D+1)=""LEFT$(STR$(X(D+1)),7)
5620 NEXT
5640 PRINTAT(6,17)"WERTE IN ORDNUNG ?"
5650 GETW$:IFW$=""THEN5650
5652 IFLEFT$(W$,1)<"J"ANDLEFT$(W$,1)<"N"THEN5640
5655 PRINTAT(6,17)W$
5660 IFLEFT$(W$,1)=""THENPRINTAT(6,17)"
5700 GOTO8000
5720 PRINTAT(6,19)"MITTELWERT = ";MW
5730 PRINTAT(6,20)"SUMME = ";MB
5800 PRINTAT(6,22)"WERTETABELLE AUSDRUCKEN"
5830 INPUT"J"
5860 IFLEFT$(V$,1)<"J"ANDLEFT$(V$,1)<"N"THEN5800
5880 IFLEFT$(V$,1)=""THENHRCOPY
5900 IFLEFT$(V$,1)=""THEN5910
5910 PRINTAT(6,23)"WERTETABELLE SPEICHERN"
5930 INPUT"J"
5960 IFLEFT$(S$,1)<"J"ANDLEFT$(S$,1)<"N"THEN5910
5980 IFLEFT$(S$,1)=""THENSCRSV2,8,2,"@":"+T$+",S,W":GOSUB9000:RETURN
6000 IFLEFT$(S$,1)=""THENRETURN
7999 END
8000 REM*****
8030 REM***** SUMME UND MITTELWERT ***
8060 REM*****
8120 MB=0
8125 FORD=1TOA:MC=MC+ABS(X(D)):NEXT
8130 FORD=1TOA:MB=MB+(X(D)):NEXT
8180 MW=MB/A
8190 MW=MW*1000:MW=INT(MW):MW=MW/1000
8195 MB=MB*1000:MB=INT(MB):MB=MB/1000
8197 MC=MC*1000:MC=INT(MC):MC=MC/1000
8200 IFM$=""THEN5800
8210 GOTO5720
8300 END
9000 REM *** FEHLERKANAL *****
9010 OPEN15,8,15
9020 INPUT#15,AA,BB$,CC,DD
9025 IFBB=0THENCLOSE15:RETURN
9030 PRINT"J";AA,BB$,CC,DD
9040 GETA$:IFA$=""THEN9040
9045 CLOSE15
9050 RETURN
9090 REM *****
9100 REM *** LETZTE GRAPHIK *****
9110 REM *****
9120 CSET 2
9140 GETZ$:IFZ$=""THEN9140
9141 IFMC=0THEN9160
9142 IFZ$<"M"ANDZ$<"B"ANDZ$<"D"ANDZ$<"C"ANDZ$<"K"THEN9140
9144 IFZ$="M"THENCSET0:GOTO220
9145 IFZ$="D"THENGOTO4000
9148 IFZ$="B"THENGOTO2000
9150 IFZ$="K"THENGOTO1000
9152 IFZ$="C"THENCOPY:GOTO9140
9160 CSET0:GOTO220
9180 END

```

Listing für die grafische Darstellung von Zahlen (Schluß)

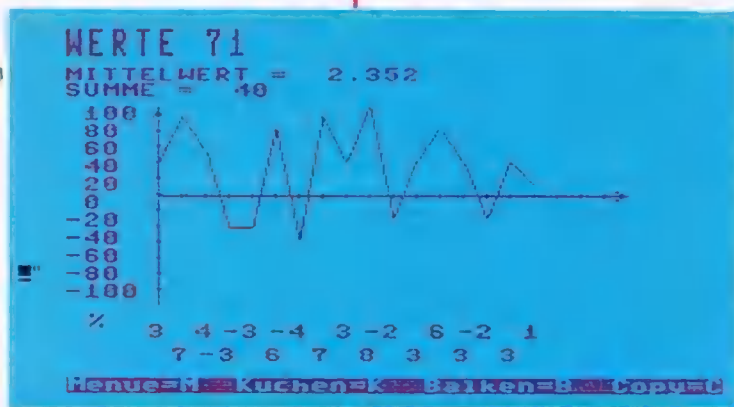


Bild 4. Werteschwankungen lassen sich mit dieser Diagrammform am besten aufzeigen

Druckers in geeigneter Weise abgeändert werden.

Farbwahl

Zur Farbwahl muß gesagt werden, daß sie — es wird ja in Hires-Grafik gearbeitet — ein wenig eingeschränkt ist. Die Auswahl muß in folgender Form durchgeführt werden: Hires Vordergrund (VG), Hintergrund (HG) = > Low col VG neu, HG, HG = < col = < Low col VG neu, HG, HG und so weiter.

Dabei wird die neue Vordergrundfarbe bestimmt. Man muß jedoch trotzdem noch aufpassen: Wenn Zeichen verschiedener Farben zu dicht aneinanderkommen oder sich überlagern, so führt das zu Farbvermischungen oder völlig anderen Farben.

Eine Ausführung im weniger problematischen Multi-Modus erwies sich jedoch wegen der Zeichengröße als nicht durchführbar.

Das Programm läuft ohne Änderungen auf dem Commodore 64 mit der Basic-Erweiterung »Simons Basic«, der Floppy 1541 und dem Epson RX-80. Falls die Datasette verwendet werden sollte, muß der SCRSV-Befehl folgendermaßen abgeändert werden SCRSV 2,8,2...—SCRSV 1,1,1.

Ansonsten dürfte das Programm mit kleinen Änderungen auch mit jedem anderen grafikfähigen Drucker laufen. (Andreas Funk)

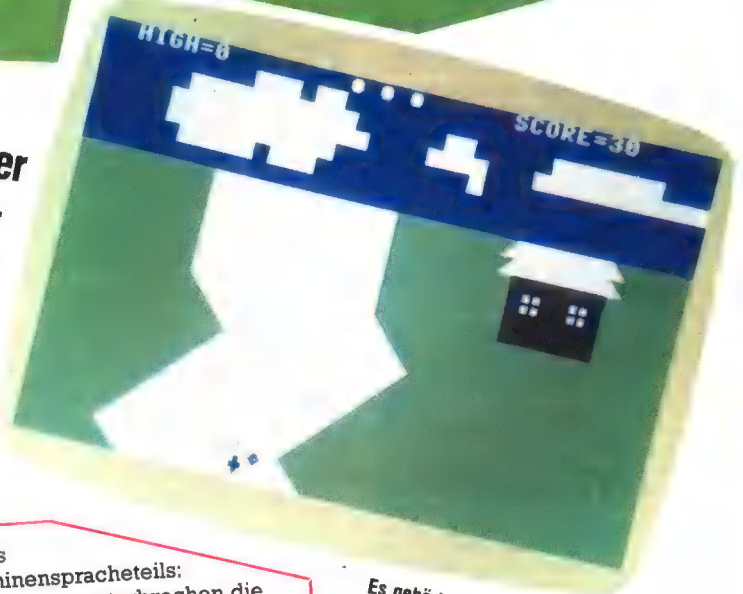
HOT WHEELS

Brausen Sie mit dem »Feuerstuhl« über die Landstraße. Das Einlegen des Ganges ist einem richtigen Motorrad nachempfunden. Ein schnelles Spiel für den Commodore 64.

Ziel des Spiels ist es, mit einem Motorrad möglichst unfallfrei eine Straße entlang zu fahren und dabei so weit wie möglich zu kommen. Dies wird durch Steine auf der Fahrbahn und Straßensperren erschwert. Nach Erreichen von 2000 Punkten gelangt man in die nächste Schwierigkeitsstufe, die dann schon etwas mehr Geschicklichkeit verlangt. Die Anzahl der Hindernisse und die Geschwindigkeit steigen von Spielstufe zu Spielstufe, wobei maximal drei Stufen durchfahren werden können.

Ein Bewegen des Joysticks in Richtung »vorwärts« erhöht die Geschwindigkeit, die bei Richtung »zurück« wieder gedrosselt wird. Der Punktestand steigt natürlich proportional zur Geschwindigkeit. Eine einprogrammierte Mindestgeschwindigkeit verhindert dabei das Aufkommen von Langeweile.

Der Kern des Spiels ist vollständig in Maschinensprache programmiert, wodurch eine Spielgeschwindigkeit erzielt wird, die mit Basic nicht zu erreichen wäre. Lediglich Start und Ende einer Spielstufe sind in Basic geschrieben.



- | | |
|-----------|--|
| 1000-1040 | Einlesen des Sprites |
| 1050-1580 | Einlesen des Maschinenspracheteils: |
| \$C000 | Hauptschleife, hier wird ununterbrochen die Straße erzeugt. |
| \$C003 | Bewirkt einen Scroll-Down der unteren Bildschirmhälfte. |
| \$C009 | Hier wird der Punktestand proportional zur Geschwindigkeit erhöht und dann ausgedruckt. Wenn die Score-Grenze für den nächsten Level erreicht ist, wird ins Basic-Programm zurückgesprungen. |
| \$C079 | Hier wird das Motorrad entsprechend der Bewegung der Joysticks gesteuert, ebenso die Frequenzen des Motorgeräusches. |
| \$C063 | Straße einen Schritt geradeaus. |
| \$C120 | Straße einen Schritt links. |
| \$C147 | Straße einen Schritt rechts. |
| \$C168 | Hier wird das Haus ausgedruckt, das immer zufällig erscheint. |
| \$C30D | Verzögerung. |
| \$C328 | Erzeugen verschiedener Hindernisse mit Zufallsgenerator im SID. |
| 1590-1600 | Einlesen der Schlußmelodie. |
| 1610-1860 | Kaltstart. |
| 1870-2080 | Warmstart. |
| 2090 | Einsprung ins Maschinenprogramm. |
| 2100-2130 | Test auf Kollision oder nächsten Level. |
| 2140-2220 | Initialisierung des SID. |
| ab 2230 | Spielende mit Schlußmelodie, Laufschrift bei Highscore, Sprung an den Programmanfang. |

Die einzelnen Programmteile von »Hot Wheels«

Es gehört schon viel Geschick dazu, in den höheren Spielstufen alle Kurven sauber zu nehmen. Wenn dann auch noch ab und zu Hindernisse auf der Fahrbahn auftauchen, kann man meistens nur noch auf die Qualität seines Sturzhelms vertrauen.

Während des Spiels ist der Bildschirm in zwei Teile geteilt; der obere Teil stellt den Horizont sowie den Spielstand dar, während im unteren Teil das Bild ständig durch das Maschinenprogramm nach unten gescrollt wird. Der Verlauf der Straße wird durch einen Zufallszahlengenerator bestimmt, ebenso das Auftreten der verschiedenen Hindernisse. Doch damit genug der Vorrede. Sie können das Programm jetzt eintippen, aber vergewissern Sie sich vor dem Starten bitte vom ordnungsgemäßen Sitz Ihres Sturzhelms!

(Bernd Günther)

832 Farbe des Straßenrandes
 833 Momentane Position der linken Straßenecke
 834 Zwischenspeicher für Joystickabfrage
 835 Flag für Feuerknopf (betätigt bei »0«)
 836 Joystick rechts
 837 Joystick links
 838 Joystick unten
 839 Joystick oben
 841-848 Zwischenspeicher für Scroll-Routine
 850-852 Zwischenspeicher für Tonfrequenzen
 853 Zähler für Hauptschleife
 855 Zähler für Haus-Grafik
 860-862 Zwischenspeicher für Verzögerungszähler
 870,871 Score-Grenze für nächsten Level
 872 Häufigkeit für das Auftreten von Hindernissen
 880,881 Momentaner Punktestand

Speicherbelegung des Maschinensprachelements

A1(), A2() Low Byte und High Byte der Noten der Schluß-
 V melodie
 SI Startadresse des VIC
 LE Startadresse des SID
 MR Momentane Anzahl der Motorräder
 HI Momentaner Level
 A\$ Highscore
 I, J, K Text für Laufschrift oder Tastaturabfrage
 Laufvariablen

Variablenliste des Basicprogramms

```

1000 DATA0,16,0,0,16,0,0,16,0,0,32,0,1,169,0,4
1010 DATA168,64,16,32,16,128,32,8,0,48,0,0,48,0,0,48
1020 DATA0,0,252,0,0,252,0,0,252,0,0,48,0,0,48,0
1030 DATA0,16,0,0,16,0,0,00,0,0,00,0,0,0,0,0
1040 RESTORE:FORI=896TO959:READA:POKEI,A:NEXT
1050 DATA076,181,193,076,012,192,076,099,192,076,214,194,162,008,181,001,157
1060 DATA072,003,189,124,195,149,001,202,208,243,160,039,177,006,145,008,177
1070 DATA002,145,004,136,016,245,162,007,181,001,056,233,040,149,001,176,004
1080 DATA232,214,001,202,202,202,016,239,165,009,201,005,208,218,165,008,201
1090 DATA024,208,212,162,008,189,072,003,149,001,202,208,248,096,162,039,173
1100 DATA064,003,157,024,217,169,160,157,024,005,202,016,242,096,032,082,192
1110 DATA160,012,169,012,024,109,065,003,170,169,032,157,024,005,202,136,208
1120 DATA249,096,169,000,141,003,220,173,001,220,141,066,003,162,004,169,000
1130 DATA157,067,003,078,066,003,062,067,003,202,016,244,169,000,141,003,220
1140 DATA173,069,003,208,013,162,013,206,000,208,208,003,206,016,208,202,208
1150 DATA245,173,068,003,208,013,162,013,238,000,208,208,003,238,016,208,202
1160 DATA208,245,173,071,003,208,035,173,001,208,201,130,144,028,206,092,003
1170 DATA162,011,206,001,208,202,208,250,238,082,003,024,173,083,003,105,020
1180 DATA141,083,003,144,003,238,084,003,173,070,003,208,035,173,001,208,201
1190 DATA221,176,028,238,092,003,162,011,238,001,208,202,208,250,206,082,003
1200 DATA056,173,083,003,233,020,141,083,003,176,003,206,084,003,173,082,003
1210 DATA141,001,212,173,083,003,141,007,212,173,084,003,141,008,212,096,238
1220 DATA065,003,032,082,192,169,012,024,109,065,003,170,169,095,157,024,005
1230 DATA202,160,011,169,032,157,024,005,202,136,208,249,169,223,157,024,005
1240 DATA206,065,003,096,032,082,192,169,012,024,109,065,003,170,169,233,157
1250 DATA024,005,202,160,011,169,032,157,024,005,202,136,208,249,169,105,157
1260 DATA024,005,096,169,007,141,087,003,162,000,160,000,138,072,152,072,032
1270 DATA012,192,173,031,208,240,004,104,076,100,194,032,121,192,173,031,208
1280 DATA240,004,104,076,100,194,032,099,192,032,013,195,104,168,104,170,189
1290 DATA102,194,153,052,005,232,189,102,194,153,052,217,232,200,192,008,208
1300 DATA237,206,087,003,173,087,003,208,189,076,194,193,032,040,195,173,065
1310 DATA003,201,008,208,003,076,104,193,234,234,234,172,027,212,240,251,192
1320 DATA010,176,247,174,027,212,224,085,176,050,140,085,003,173,065,003,056
1330 DATA237,085,003,144,237,152,072,032,214,194,032,012,192,173,031,208,208
1340 DATA118,032,121,192,173,031,208,208,110,206,065,003,032,032,193,032,013
1350 DATA195,104,168,136,208,221,076,181,193,224,170,176,035,152,072,032,214
1360 DATA194,032,012,192,173,031,208,208,076,032,121,192,173,031,208,208,068
1370 DATA032,099,192,032,013,195,104,168,136,208,224,076,181,193,140,085,003
1380 DATA173,065,003,024,109,085,003,176,148,201,026,176,144,152,072,032,214

```

Listing von »Hot Wheels«

Listing von »Hot Wheels« (Fortsetzung)

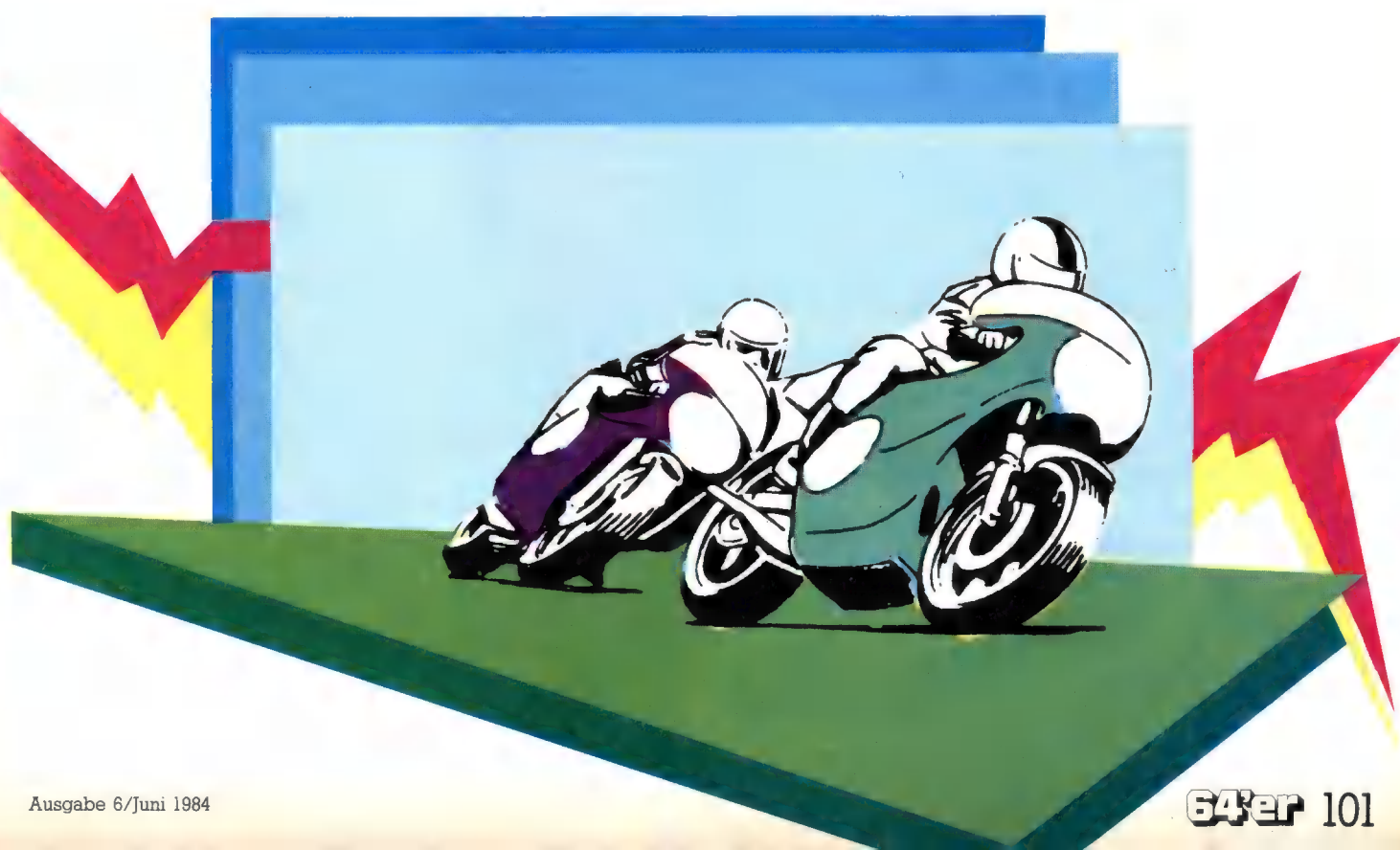

```

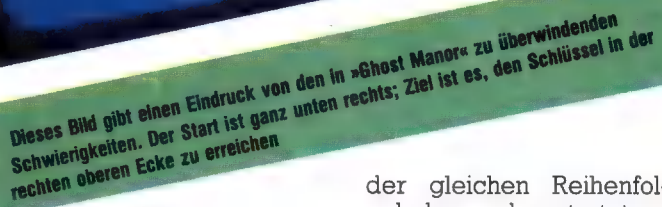
2030 GOSUB2140
2040 SYS49158
2050 FORI=1TO18:SYS49155:NEXT
2060 PRINT"XXXXXXXXXXXXLEVEL"LE
2070 FORI=1TO18:SYS49155:FORJ=1TO100:NEXT:NEXT
2080 POKEV+21,1
2090 PRINT"X";SYS49152
2100 IFPEEK(880)+256*PEEK(881)<LE*2000ORLE=3THENGOSUB2200:MR=MR-1:GOTO1870
2110 IFLE<3THENLE=LE+1:IFLE<3THEN1870
2120 POKE870,255:POKE871,255:GOTO1870
2130 END
2140 POKESI,0:POKESI+1,3:POKE850,3:POKESI+4,0
2150 POKESI+5,255:POKESI+6,255
2160 POKESI+7,80:POKE851,80:POKESI+8,0:POKE852,0:POKESI+11,0
2170 POKESI+12,255:POKESI+13,255
2180 POKESI+4,33:POKESI+11,129
2190 POKESI+24,138:RETURN
2200 POKESI+4,0:POKESI+11,0:POKEV+21,0:POKESI+1,50:POKESI+5,0
2210 POKESI+6,250:POKESI+4,129:POKESI+4,128
2220 FORI=1TO200:POKEV+21,0:POKEV+21,1:NEXTI:RETURN
2230 POKE53280,13:POKESI+11,0:POKEV+21,0:PRINT"X"
2240 FORI=1TO9:POKESI,A1(I):POKESI+1,A2(I):POKESI+4,17
2250 :FORJ=1TO200:NEXT:POKESI+4,16:NEXT
2260 IFPEEK(880)+256*PEEK(881)<HITHEN1000
2270 HI=PEEK(880)+256*PEEK(881)
2280 A$=""....."
2290 A$=A$+"HIGHSCORE"+A$:POKE53281,1:PRINT"XXXXXXXXXXXXTAB(18)HI
2300 FORK=1TO3:FORI=1TO59
2310 PRINT"XXXXXXXXXXXXMID$(A$,I,40)
2320 PRINT"XXXXXXXXXXXXMID$(A$,60-I,40)
2330 FORJ=1TO100:NEXT:NEXT:NEXT:GOTO1000

```

READY.

Listing von »Hot Wheels« (Schluß)





der gleichen Reihenfolge geladen und gestartet werden. Haben Sie den Ladevorgang beendet, wird das Spiel durch die Taste »0« oder durch den Aktionsknopf des Joysticks gestartet. Das Programm kann mit Joystick oder über die Tastatur gespielt werden. Ist kein Joystick vorhanden, lenken Sie Ihr Männchen mit folgenden Tasten:
A: oben; Z: unten; B: links; M: rechts; N: Sprung.

Beim Vorhandensein eines Joysticks dient der Aktionsknopf als Sprungknopf.

Listing 1. Grafik für Ghost Manor

MANOR

Variablenliste zu »Ghost Manor«

A Zeiteinheiten
B,C Koordinaten des Männchens
S Sektornummer
F In Reserve befindliche Leben
H,I,G,T Laufvariablen
M Zeigt an, ob Männchen gerade springt
R Punktzahl
Q,W Zeichen hinter Männchen
P,O Zeichen für Männchen
X Speicherstelle für die nächste Note der Melodie
U Richtung für Sprung
L,K Bewegung der Beine

daß dem Männchen keine Steine auf den Kopf fallen. Hat man schließlich alle Gefahren gut überstanden, klettert man auf die Zinne und stellt sich unter den Schlüssel, wodurch das Schloßtor geöffnet wird.

Nun werden die Fallgruben übersprungen, und man steht vor dem letzten großen Hindernis, der Brücke. Diese Brücke schließt und öffnet sich periodisch, und es gilt, den richtigen Moment abzu-

```

1 REM GHOST MANOR
2 REM MASCHINENU.P.
3 REM BY FRANK BACKES
10 FORI=67000TO7110:READY:POKEI,Y:NEXT
20 DATA160,16,185,206,31,201,15,240,15,201,16,240,26,136,192,1,240,3,24,144,237,
24,144
21 DATA30,169,32,153,206,31,169,16,200,153,206,31,136,24,144,230,169,32,153,206,
31,169
22 DATA15,200,153,206,31,136,24,144,215,173,223,31,201,15,240,3,24,144,10,169,32,
,141
23 DATA223,31,169,16,141,208,31
24 DATA160,2,185,118,31,201,18,240,15,201,19,240,26,200,192,17,240,3,24,144,237,
24,144
25 DATA30,169,32,153,118,31,169,19,136,153,118,31,200,24,144,230,169,32,153,118,
31,169
26 DATA18,136,153,118,31,200,24,144,215,173,119,31,201,18,240,3,24,144,10,169,32,
,141
27 DATA119,31,169,19,141,134,31
28 DATA160,132,185,154,30,201,21,240,8,136,192,0,240,26,24,144,241,169,32,153,15,
4,30
29 DATA152,24,105,22,168,169,21,153,154,30,152,24,233,22,168,24,144,225,160,13,1
85,30,31
30 DATA201,21,240,8,200,192,22,240,16,24,144,241,169,32,153,30,31,169,21,153,154
,30,24
31 DATA144,235
32 DATA160,13,185,110,30,201,23,240,8,200,192,19,240,18,24,144,241,169,32,153,11
92,11
33 DATA169,23,136,153,110,30,200,24,144,233,160,17,185,110,30,201,22,240,8,136,1
92,11
34 DATA240,18,24,144,241,169,32,153,110,30,169,22,200,153,110,30,136,24,144,233
35 DATA173,122,30,201,23,240,10,173,128,30,201,22,240,16,24,144,23,169,32,141,12
2,30
36 DATA169,22,141,122,30,24,144,10,169,32,141,128,30,169,23,141,128,30
37 DATA173,223,30,201,30,240,10,173,229,30,201,31,240,16,24,144,18,169,31,141,22
3,30
38 DATA169,31,141,223,30,24,144,5,169,30,141,229,30,160,8,185,220,30,201,31,240,
8,136
39 DATA192,2,240,18,24,144,241,169,32,153,220,30,200,169,31,153,220,30,136,24,14
4,21
40 DATA160,4,185,220,30,201,30,240,8,200,192,11,240,7,24,144,241
41 DATA136,153,220,30,96
READY.
```

Listing 2. Maschinenspracherroutine für Ghost Manor

Zu Beginn des Spiels befindet sich das Männchen rechts unten vor seiner Zelle. Nun kommen ihm Spinnen entgegengekrabbeln, welche es überspringen muß. In der nächsten Etage rollen ihm Fässer entgegen,

und es muß mit einem Doppelsprung eine Fallgrube überspringen. Einen Doppelsprung macht man folgendermaßen: Man drückt die Sprungtaste »N« und kurz danach die Taste für die

Richtungsangabe (B oder M). Genauso verfährt man, wenn man mit Joystick spielt: Aktionsknopf kurz drücken, danach Richtungsangabe. Ist auch dieser Teil geschafft, muß man aufpassen,

passen, um die Brücke überqueren zu können. Die letzten Fallgruben werden übersprungen, und Sie können aus dem Schloß hinauslaufen. Geschafft! Eine kurze Erholungspause, und das Spiel beginnt von vorne, natürlich etwas schwieriger.

Die Sektornummer wird links oben, die Anzahl der in Reserve verbliebenen Leben (man beginnt mit drei


```

50 POKE650,128:SYS7456:GETA$:IFA$="0"ORPEEK(252)=0THEN70
60 GOT050
70 R=0:F=3:S=1
80 B=8142:C=17:A=3000:Q=32:W=32:O=2:P=1:X=7491
100 PRINT"J":POKE36879,14:POKE36869,255:PRINT"***GHOST MANOR**"
110 PRINT"#####"
111 PRINT"Q"
112 PRINT"Q 1 1 1 QQQQQQ QQQQ"
113 PRINT"Q Q Q"
114 PRINT"Q Q U Q"
115 PRINT"Q Q Z"
116 PRINT"QQ ↑↑↑↑↑↑↑↑ Q"
117 PRINT"Q Q Q"
118 PRINT"Q Q Q"
119 PRINT"Q Q U Q"
120 PRINT"JJQ QQQ QQQQQQ QQQQQJJ";
121 PRINT"Q @ Q";
122 PRINT"Q @ S Q";
123 PRINT"Q R @SRRQ";
124 PRINT"LOKK#KKK KKK#KKKKKKKK#";
125 PRINT"LO N N TTTM";
126 PRINT"LO ATTMM";
127 PRINT"LO P BTMM";:IFS>3THENPOKE7893,21:POKE8090,32
128 PRINT"#JJJJJJJJJJJJJJJJJJJJ";:IFS>2THENPOKE7847,21:POKE8059,18
129 POKE8164+21,35:POKE7746+18,24:IFS>1THENPOKE7806,22:POKE8152,16
130 PRINT"#####TIME":A;"|| SCORE":R;"|| ":POKE36878,0:POKE36877,0:U=0:IFM=2THENM=0
131 POKE7680,S+176:POKE7701,F+175
140 POKE36878,15:POKE36876,PEEK(X):X=X+1:IFX=7507THENX=7491
150 POKEB+C,0:POKEB+C-22,P
155 IFM=1ANDPEEK(B+C+22)<32THENR=R+50
160 SYS6700:IFPEEK(B+C)<0ORPEEK(B+C-22)<0PTHEM500
165 IFPEEK(B+C+22)=32ANDM=0THEN600
170 IFB+C=7967THEN1000
180 A=A-10:IFA=-10THEN500
185 IFPEEK(B+C-44)=24THEN800
188 IFM=1THENM=2:POKEB+C,W:POKEB+C-22,Q:B=B+22:C=C+U
189 SYS7456:IFPEEK(B+C)<0ANDM=1ORPEEK(B+C)<32ANDM=2THEN500
190 GETA$:IFA$="A"ANDPEEK(251)=156ANDPEEK(252)<0THENFORT=1T050:NEXTT:GOT0130
195 POKEB+C,W:POKEB+C-22,Q:POKE36878,15:POKE36877,180
200 IFA$="A"ORPEEK(251)=152THEN250
210 IFA$="Z"ORPEEK(251)=148THEN300
220 IFA$="B"ORPEEK(251)=140THEN350
230 IFA$="M"ORPEEK(251)=28THEN400
240 IFA$="N"ANDM=0ORPEEK(252)=0ANDM=0THEN450
245 GOT0130
250 IFW=0THENB=B-22
260 IFP=7THEN0+1:IFO=10THEN0=8
265 IFP=7THEN490
270 P=7:Q=8:GOT0490
300 IFPEEK(B+C+22)=0THENB=B+22
310 IFP=7THEN0=0+1:IFO=10THEN0=8
315 IFP=7THEN490
320 P=7:Q=8:GOT0490
350 IFPEEK(B+C-1)=0ORPEEK(B+C-1)=32THENC=C-1
360 IFP=1THEN0=0+1:IFO=4THEN0=2
365 IFP=1THEN490
370 P=1:Q=2:GOT0490
400 IFPEEK(B+C+1)=0ORPEEK(B+C+1)=32THENC=C+1
410 IFP=4THEN0=0+1:IFO=7THEN0=5
415 IFP=4THEN490
420 P=4:Q=5:GOT0490
450 IFP=1THENU=1
451 IFP=4THENU=-1
455 IFPEEK(B+C-44-U)=32THENB=B-22:M=1:GOT0470
460 GOT0130
470 C=C-U
490 W=PEEK(B+C):Q=PEEK(B+C-22):GOT0130
500 FORH=1T05
501 POKEB+C,34:POKEB+C-22,33:POKE36878,15:FORG=200T0250:POKE36876,G:NEXT:POKE368
76,0
502 NEXTH
503 POKE198,0
510 POKE36878,0:F=F-1:IFF=0THEN530
520 GOT080
530 PRINT"#####GAME OVER!":FORT=1T0200:NEXT:GOT050
600 POKEB+C-22,32:B=B+22:IFPEEK(B+C)<32THENPOKEB+C,32:B=B-22:GOT0500
605 POKEB+C,0:POKEB+C-22,P
610 FORT=1T0100:NEXT:GOT0600
800 POKE7968,32:FORT=1T0150:NEXT:POKE7946,32:FORT=1T0150:NEXT:POKE7924,32
810 POKEB+C-44,32:POKE7768+C,32:POKE7790+C,32:GOT0490
1000 POKE36876,0:POKE36878,15:FORG=1T010:FORH=150T0250STEP2:POKE36875,H:NEXTH,G
1010 POKE36875,0:POKE36878,0:S=S+1:R=R+A:IFS=2THENF=F+1
1020 GOT080

```

READY.

Leben) wird rechts oben angezeigt. Für jeden Versuch hat man 3000 Zeiteinheiten zur Verfügung. Hat man einen Sektor geschafft, bekommt man die verbliebenen Zeiteinheiten als Bonus gutgeschrieben. Nach dem ersten Sektor erhält man ein Extra-Leben, und es taucht ein Gespenst auf, welches den Schlüssel auf der Zinne bewacht. Noch ein Tip: Wenn Sie vor einem aus der Decke ragenden Felsen stehen, können Sie nicht springen.

Nutzt die grafischen Möglichkeiten des VC 20 voll aus

Das Spiel ist nicht ganz einfach, man braucht schon ein bißchen Übung, um den ersten Sektor zu schaffen. Leider ist das Spiel nur in Schwarzweiß programmiert, aber eine farbliche Gestaltung des Bildschirms war mit dem beschränkten Speicherplatz der VC 20-Grundversion nicht mehr möglich.

Als Ausgleich werden die grafischen Fähigkeiten des VC 20 voll ausgenutzt. Alle Bewegungen von Figuren werden aus Geschwindigkeitsgründen von einem Maschinenspracheprogramm ausgeführt.

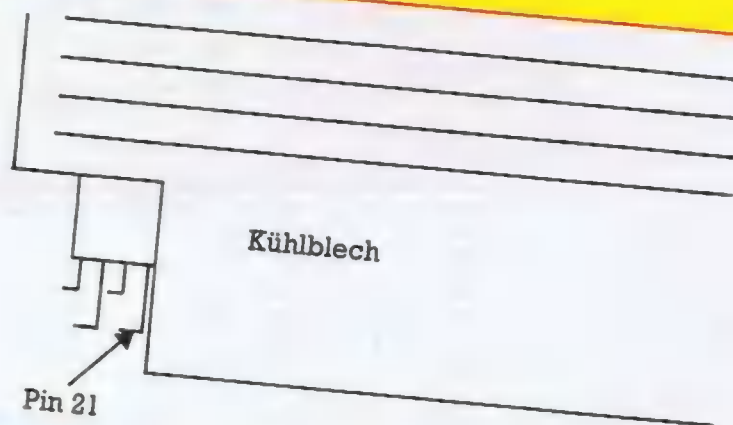
(Frank Backes)

Listing 3. Ghost Manor — Das Hauptprogramm

Steckmodule auf Kassette

Vielleicht kennen Sie das Problem. Sie haben einen VC 20, eine Vollerweiterung und ein Programmierhilfemodul, aber keine Modulbox. Sie können das Programmierhilfemodul nur in der Grundversion einsetzen, möchten aber auch gerne bei 16 KByte damit arbeiten. Hier wird nun die Methode vorgestellt, das Modul auf Kassette zu kopieren und dann bei Vollerweiterung in den jeweiligen Speicherbereich zu laden.

Dieses Bild zeigt, wo Sie am geöffneten VC 20 eingreifen können, um einen Modulschalter anzubringen.



Anhand der SYS-Adresse, mit der Sie das Programm starten, sehen Sie, in welchem 16- oder 8-KByte-Block das Modul liegt. Zum Beispiel Start mit SYS 32592, das Modul befindet sich also im Bereich von 16384 bis 32768.

Kopieren auf Band

Stecken Sie das Modul in den Erweiterungsport und schalten Sie den VC 20 ein. Legen Sie eine Kassette in den Recorder und tippen Sie POKE 43,0:POKE 44,64:POKE 45,0:POKE 46,128:Save"(Name)",1 ein, wenn das Modul ab 16384 liegt, oder POKE 43,0:POKE 44,160:POKE 45,0:POKE 46,192:Save"(Name)",1 wenn das Programm ab 40960 gespeichert ist.

Stecken Sie die für das Modul benötigte 16- oder 8-KByte-Erweiterung in den Port, legen Sie die Kassette ein und laden Sie das Programm mit »Load«. So einfach ist das. Nach dem Laden starten Sie das Modul mit der gewohnten SYS-Adresse.

Der Autostart bei Modulen ...

Module, die im Bereich von 40960 liegen, werden beim Einschalten selbst gestartet, wenn die Anfangsadressen folgendermaßen belegt sind:

40960	low	Startadresse des Programms	
40961	high		
40962	low	Adresse bei Restore	
40963	high		
40964	65	= a	
40965	48	= 0	
40966	195	= C	
40967	194	= B	
40968	205	= M	

Während eines Kaltstartes werden die Adressen von 40964 bis 40968 nach der Zeichenfolge "a0CBM" abgelesen und das Programm gegebenenfalls gestartet.

Wenn Sie über eine Speichererweiterung in diesem Bereich verfügen, manipulieren Sie doch einmal diese Adressen, lassen Sie die Startadresse auf ein Maschinenprogramm zeigen und

tippen Sie SYS 64802 (Kaltstart) ein. Sie werden sehen, daß das Maschinenprogramm gestartet wird.

... und seine Überlistung

Da der Computer die Adressen für den Autostart nur beim Kaltstart absucht, könnte man den Computer einschalten und dann das Modul einstecken. Mit viel Glück könnte es klappen, aber es können auch erhebliche Schäden am Computer oder am Modul auftreten.

Softwaremäßig gibt es keine Lösung, den Autostart zu verhindern. Ein kleiner Eingriff in die Hardware macht's aber doch möglich. Auf Seite 150 des VC 20-Handbuches ist die Hauptspeicher-Erweiterung beschrieben. Uns interessiert Pin 21 (+ 5 V). Wenn Sie Ihren VC 20 öffnen, sehen Sie die Anordnung entsprechend dem Bild.

Kneifen Sie den Draht von Pin 21 durch und löten Sie zwei nicht zu dünne, isolierte Kabel an. Die Kabel verbinden Sie mit einem Schalter, den Sie im Gehäuse einbauen oder nach außen legen können. Mittels dieses Schalters können Sie Module ein- und ausschalten!

Also, wir stecken das Modul bei ausgeschaltetem Computer in den Port und unterbrechen Pin 21 mit unserem Schalter. Dann schalten wir den VC 20 ein, der Computer findet in den betreffenden Adressen kein a0CBM und meldet sich mit dem üblichen Bild. Jetzt drücken wir unseren Schalter, der Strom fließt, aber es wird kein Autostart mehr ausgeführt. Wir können das Modul jetzt mit POKE 43,0:POKE 44,160:POKE 45,0:POKE 46,192:SAVE"(Name)",1 auf Kassette abspeichern oder mit SYS 64802 (Kaltstart, wir erinnern uns) starten.

(Jörg Grohne)

Logic Disassembler

Wer hat nicht schon nach einem guten Einstieg in das Betriebssystem oder in ein selbstgeschriebenes Assembler-Programm gesucht, um dem Verlauf der Programmlogik folgen zu können?

Mit diesem ganz in Basic geschriebenen Programm besteht nun die Möglichkeit, bei einer beliebigen Startadresse einzusteigen und von dort aus das Programm zu verfolgen.

Auch mit einem kompletten Disassembler-Listing war dies bisher eine äußerst aufwendige und zeitraubende Angelegenheit. Bei jedem Sprung-Befehl (sei es BRANCH, JSR oder JMP) ging das große Blättern los. Das Verfolgen von mehrstufigen Unterprogrammen mit unterschiedlichen Ausstiegsmöglichkeiten ist schon ganz unmöglich.

Der Logic-Disassembler hingegen wartet bei jedem Sprungbefehl auf die Eingabe von Y, N oder X. Y steht für »yes«, das heißt dem Sprungbefehl folgen, N steht für »no«, also zum Beispiel eine bekannte Unterroutine einfach zu übergehen, damit das Listing nicht unübersichtlich wird. X schließlich steht für »exit«, wodurch der Disassembler bei einer neuen Adresse gestartet werden kann.

Das Listing der Basic-Routine »NEW« (Bild 1) zeigt als Beispiel, wie ein solcher Disassembler-Lauf vor sich geht.

Die Einsprungsadresse liegt bei \$C642.

In Zeile 1 wird der BNE-Befehl mit der Eingabe von »N« übergangen. Der JSR-Befehl in Zeile 14 wird mit »Y« verfolgt, und es erfolgt eine Verzweigung in die Routine \$C68E. Mit der Anzeige von STACK 0 wird über die Tiefe der Unterprogramm-Verschachtelung informiert. Bei Erreichen von RTS in Zeile 22 wird in das Hauptprogramm (STACK 0) zurückgesprungen und dieses ab der Adresse \$C65C fortgeführt.

Der darauffolgende BRANCH-Befehl in Zeile 24 wird wieder mit der Eingabe von »N« übergangen. In Zeile 25 folgt dann ein RTS, und da der STACK auf 0 steht (man befindet sich also im Hauptprogramm), wird der Disassembler-Vorgang beendet.

<LOGIC - DISASSEMBLER>

LINE#	LOC	CODE	STATEMENT	
	0000		#\$C642	
0001	C642	D0 FD	BNE \$C641	
0002	C644	A9 00	LDA #00	
0003	C646	A8	TAY	
0004	C647	91	STA (\$2B),Y	
0005	C649	C8	INY	
0006	C64A	91 2B	STA (\$2B),Y	
0007	C64C	A5 2B	LDA \$2B	
0008	C64E	18	CLC	
0009	C64F	69 02	ADC #02	
0010	C651	85 2D	STA \$2D	
0011	C653	A5 2C	LDA \$2C	
0012	C655	69 00	ADC #00	
0013	C657	85 2E	STA \$2E	
0014	C659	20 8E C6	JSR \$C68E	STACK 1
0015	C68E	18	CLC	
0016	C68F	A5 2B	LDA \$2B	
0017	C691	69 FF	ADC #FF	
0018	C693	85 7A	STA \$7A	
0019	C695	A5 2C	LDA \$2C	
0020	C697	69 FF	ADC #FF	
0021	C699	85 7B	STA \$7B	
0022	C69B	60	RTS	STACK 0
0023	C65C	A9 00	LDA #00	
0024	C65E	D0 2D	BNE \$C66D	
0025	C66D	60	RTS	

START-ADR. = \$C642
END-ADR. = \$C66D

< DISASSEMBLER - END >

Bild 1.
Logische Disassemblierung
der Basic-Routine »NEW«

```

100 POKE36879,233:PRINT"*** DISASSEMBLER ***"
110 PRINT"LOGIC - DISASSEMBLER"
120 PRINT"VC 20 - VERSION"
130 PRINT"BY "
140 PRINT"FRED HAMMER"
150 PRINT"OBERE BERGSTR. 12"
160 PRINT"5419 LEUTEROD"
170 PRINT"TEL. 02602/60372"
180 PRINT"PRESS ANY KEY"
190 GETA$:IF A$="" THEN 190
200 CLR:POKE36879,25:DIM S$(100):SS$=0:DIM AC$(255)
210 PRINT"X DISASSEMBLER-START"
220 FOR I=0 TO 255:READ AC$(I):NEXT I
230 POKE36879,25:PRINT"X DISASSEMBLER-START"
240 H$="0123456789ABCDEF"
250 GOT0520
260 REM HEX-UMRECHNUNG
270 DE=0:FOR Z0=1 TO 4:N0=0:FOR Z1=1 TO 16
280 IF MID$(X$,Z0,1)=MID$(H$,Z1,1) THEN N0=Z1-1:Z1=16
290 NEXT Z1:DE=DE+16*(4-Z0)*N0:NEXT Z0:RETURN
300 REM DEZ-UMRECHNUNG
310 X$="":FOR Z0=4 TO 1 STEP -1:Z1=INT(DE/16*(Z0-1))
320 IF Z1=0 THEN X$=X$+"0":GOTO 350
330 IF Z1<10 THEN X$=X$+RIGHT$(STR$(Z1),LEN(STR$(Z1))-1):GOTO 350
340 X$=X$+CHR$(Z1+55)
350 DE=DE-Z1*16*(Z0-1):NEXT Z0:RETURN
360 REM DER 6502-BEFEHLSSTZ
370 DATA BRK,ORA (X,,,ORA $,ASL $,,PHP,ORA #,ASL,,,ORA,ASL?,BPL,ORA (Y,,,
380 DATA ORA $X,ASL $X,,CLC,ORA Y,,,ORA X,ASL X,,JSR?,AND (X,,,BIT $,AND $
390 DATA ROL $,,PLP,AND #,ROL,,BIT?,AND?,ROL?,BMI,AND (Y,,,AND $X,ROL $X,
400 DATA SEC,AND Y,,,AND X,ROL X,,RTI,EOR (X,,,EOR $,LSR $,,PHA,EOR #
410 DATA LSR,,JMP?,EOR?,LSR?,BVC,EOR (Y,,,EOR $X,LSR $X,,CLI,EOR Y,,,EOR X
420 DATA LSR X,,RTS,ADC (X,,,ADC $,ROL $,,PLA,ADC #,ROL,,JMP (,ADC?,ROR?,
430 DATA BVS,ADC (Y,,,ADC $X,ROR $X,,SEI,ADC Y,,,ADC X,ROR X,,STA (X,,
440 DATA STY $,STA $,STX $,,DEY,,TXA,,STY?,STA?,STX?,BCC,STA (Y,,,STY $X
450 DATA STA $X,STX $Y,,TYA,STA Y,,TXS,,STA X,,LDY #,LDX #,,LDY $X
460 DATA LDA $,LDX $,,TAY,LDA #,TAX,,LDY?,LDA?,LDX?,BCS,LDA (Y,,,LDY $X
470 DATA LDA $X,LDX $Y,,CLV,LDA Y,,TXS,,LDY X,,LDA X,,LDX Y,,CPY #,CMP (X,,
480 DATA CPY $,CMP $,DEC $,,INY,CMP #,DEX,,CPY?,CMP?,DEC?,BNE,CMP (Y,,,
490 DATA CMP $X,DEC $X,,CLD,CMP Y,,,CMP X,DEC X,,CPX #,SBC (X,,CPX $
500 DATA SBC $,INC $,,INX,SBC #,NOP,,CPX?,SBC?,INC?,BEQ,SBC (Y,,,SBC $X
510 DATA INC $X,,SED,SBC Y,,,SBC X,INC X,
520 FE$=ADS:INPUT"START (HEX)";A$
530 IF A$="" THEN CLR:POKE36879,29:END
540 IF LEN(A$)<>5 OR LEFT$(A$,1)<>"$" THEN 520
550 X$=RIGHT$(A$,4):GOSUB 260:A=DE
560 PRINT"XPRINTER ? Y/N":POKE198,0:WAIT 198,1:GETAN$:IF AN$<>"Y" THEN 600
570 INPUT"COMMENT ";FF$
580 OPEN 4,4:CMD 4
590 PRINT CHR$(14);FF$:CHR$(15):PRINT
600 PRINT"XLOGIC - DISASSEMBLER":PRINT:PRINT"LINE# LOC CODE STATE"
610 PRINT:PRINT" 0000"
620 AD=A-1
630 AD=AD+1:N=N+1:DE=AD:GOSUB 300
640 PRINT RIGHT$("0000"+RIGHT$(STR$(N),LEN(STR$(N))-1),4) "X$ "
650 P=PEEK(AD):DE=P:EA=P:GOSUB 300:S$=RIGHT$(X$,2):RESTORE
660 MN$=AC$(P):IF LEN(MN$)=3 THEN 800
670 IF MN$="" THEN 940
680 IF LEN(MN$)=4 THEN 830
690 IF RIGHT$(MN$,1)="#" THEN MN$=MN$+"$":GOTO 770

```

Das Listing zum »Logic Disassembler«

<LOGIC - DISASSEMBLER>

LINE# LOC CODE STATEMENT

```

0000      0000      **=$FFE1
0001 FFE1 6C 03 28 JMP ($0328)
0002 F770 A5 91 LDA $91
0003 F772 C9 FE CMP #$FE
0004 F774 D0 07 BNE $F77D
0005 F776 08 PHP
0006 F777 20 CC FF JSR $FFCC STACK 1
0007 FFCC 6C 03 22 JMP ($0322)
0008 F3F3 A2 03 LDX #03
0009 F3F5 E4 9A CPX $9A
0010 F3F7 30 03 BCS $F3FC
0011 F3F9 20 04 EF JSR $EF04 STACK 2
0012 F3FC E4 99 CPX $99
0013 F3FE 30 03 BCS $F403
0014 F400 20 F6 EE JSR $EEF6 STACK 2
0015 F403 86 9A STX $9A
0016 F405 A9 00 LDA #00
0017 F407 85 99 STA $99
0018 F409 60 RTS STACK 3
0019 F77A 85 C6 STA $C6
0020 F77C 28 PLP
0021 F77D 60 RTS
  
```

START-ADR. = \$FFE1
END-ADR. = \$F77D

< DISASSEMBLER - END >

Bild 1.

Die Betriebssystemroutine zur Abfrage der STOP-Taste disassembliert.

```

700 IFRIGHT$(MN$,1)="$"THEN770
710 IFEA=108THEN840
720 IFMID$(MN$,5,1)("<"THEN790
730 IFMID$(MN$,5,1)="$"THEN810
740 GOSUB750:MN$=LEFT$(MN$,4)+"$"+AS$(2)+AS$(1)+", "+RIGHT$(MN$,1):GOTO940
750 FORK=1TO2:AD=AD+1:P=PEEK(AD):DE=P:GOSUB300:AS$(K)=RIGHT$(X$,2)
760 S$=S$+" "+RIGHT$(X$,2):NEXT:RETURN
770 AD=AD+1:P=PEEK(AD):DE=P:GOSUB300:S$=S$+" "+RIGHT$(X$,2)
780 MN$=MN$+RIGHT$(X$,2):GOTO940
790 AD=AD+1:P=PEEK(AD):DE=P:GOSUB300:S$=S$+" "+RIGHT$(X$,2)
800 MN$=LEFT$(MN$,5)+"$"+RIGHT$(X$,2)+", "+RIGHT$(MN$,1):GOTO940
810 AD=AD+1:P=PEEK(AD):DE=P:GOSUB300:S$=S$+" "+RIGHT$(X$,2)
820 MN$=LEFT$(MN$,5)+RIGHT$(X$,2)+", "+RIGHT$(MN$,1):GOTO940
830 GOSUB750:MN$=LEFT$(MN$,3)+"$"+AS$(2)+AS$(1):GOTO940
840 AD=AD+2:P=PEEK(AD):DE=P:GOSUB300:S$=S$+" "+RIGHT$(X$,2)
850 MN$=LEFT$(MN$,5)+"$"+RIGHT$(X$,2)
860 P=PEEK(AD-1):DE=P:GOSUB300:S$=S$+" "+RIGHT$(X$,2)
870 MN$=MN$+RIGHT$(X$,2)+", ":GOTO940
880 REM BRANCH-BEFEHL
890 IFMN$="BRN"ORLEFT$(MN$,1)<"B"THEN940
900 AD=AD+1:P=PEEK(AD):DE=P:GOSUB300:S$=S$+" "+RIGHT$(X$,2)
910 IFP=130THENAD=AD-(255-P):GOTO930
920 AD=(AD-1)+P+2
930 DE=AD:GOSUB300:MN$=MN$+"$"+X$:KZ=1
940 IFEA=32THENMN$=MN$+" STACK"+STR$(SS$+1)
950 IFEA=96ANDSS$=0THENMN$=MN$+" STACK"+STR$(SS$-1)
960 PRINTLEFT$(S$+" ",16):MN$:IFEA=96THENPRINT
970 IFKZ=0THEN1030
980 GETAA$:IFAA$=""THEN980
990 IFAN$="Y"THENCMD4,CHR$(15);
1000 KZ=0:IFAA$="N"THEN1030
1010 IFAA$="X"GOTO1260
1020 PRINT"AD=AD-1 KZ=0"
1030 IFEA=32OREA=76OREA=108OREA=96THEN1050
1040 GOTO630
1050 IFEA=96THEN1090
1060 REM "BEFEHL: RTS"
1070 IFSS$=0THEN1260
1080 SS$=SS$-1:AD=SS$(SS$):GOTO630
1090 REM "BEFEHL: JSR"
1100 GETAA$:IFAA$=""THEN1100
1110 IFAN$="Y"THENCMD4,CHR$(15);
1120 IFAA$="N"THEN630
1130 IFAA$="X"GOTO1260
1140 IFEA=32THEN1180
1150 SS$(SS$)=AD:SS$=SS$+1:PRINT
1160 AD=PEEK(AD)*256+PEEK(AD-1)-1
1170 GOTO630
1180 REM "BEFEHL: JMP ABS."
1190 IFEA=76THEN1210
1200 AD=PEEK(AD)*256+PEEK(AD-1)-1:PRINT:GOTO630
1210 REM "BEFEHL: JMP IND."
1220 A1=PEEK(AD)*256+PEEK(AD-1)
1230 A2=PEEK(AD)*256+PEEK(AD-1)
1240 IFA1>A2THENPRINT:PRINT"LOGIC FLOW ERROR":GOTO1260
1250 AD=PEEK(A1+1)*256+PEEK(A1)-1:PRINT:GOTO630
1260 DE=AD:GOSUB300
1270 PRINT:PRINT"START-ADR. = "AS$:PRINT"END-ADR. = "X$
1280 PRINT:PRINT"< DISASSEMBLER - END >"
1290 IFAN$="Y"THENPRINT#4,:CLOSE4
1300 GOTO520
  
```

Listing zum »Logic Disassembler«
(Schluß)

As - Tabelle fuer Sprungadressen
ACX - Tabelle fuer 6502-Mnemonics
SSX - Tabelle fuer Stack
I - Index fuer Stack
H\$ - Index fuer Mnemonic-Tabelle
FES - Werte fuer Hexa-Umrechnung
AS - Zwischenspeicher fuer Anfangsadresse
X\$ - Startadresse in Hexa
DE - Hexa-Wert bei Umrechnung
Z0] - Dezimal-Wert bei Umrechnung
Z1] - Indexfelder fuer Umrechnung
A - Anfangsadresse in Dezimal
AN\$ - KZ fuer Druckerausgabe
N - Zeilennummer
AD - aktuelle Adresse bei der Disassemblierung
P - Zwischenspeicher fuer Adressrechnung
EA - Dezimal-Wert des laufenden Befehls
S\$ - Hexa-Werte der laufenden Disassembler-Zeile
MN\$ - laufende Disassembler-Zeile
K - Anzahl der Bytes fuer das Befehlswort
AAs - Eingabefeld bei Sprungbefehlen
FF\$ - Überschrift bei Druckerausgabe
A1 - Arbeitsfeld zur Adressberechnung bei JMP ind.
A2 - wie A1, jedoch zur Kontrolle ob Memory vorhanden

Variablen-Liste vom
»Logic Disassembler«

und auf die erneute Eingabe einer Startadresse gewartet. Bei der Eingabe von »END« statt einer Startadresse wird das Programm beendet.

Dabei wird jede Stackänderung angezeigt.

Es können alle Sprungbefehle verfolgt werden, sogar solche, die sich auf eine indirekte Adressierung beziehen. Als Beispiel hierfür kann das Disassembler-Protokoll der Abfrage der Stop-Taste dienen (Bild 2).

In der Zeile 1 ist der Befehl JMP (\$0328) gefunden worden, jetzt wird der Inhalt der Speicherstelle \$0328 und \$0329 ausgelesen und die Sprungadresse errechnet. Dann erfolgt ein Speicher-test an der errechneten Sprungadresse, und erst, wenn dieser Test positiv verläuft, wird ab dieser Adresse weiter disassembliert. Verläuft der Test jedoch negativ, erfolgt ein »LOGIC FLOW ERROR«, und es wird zum

100 - 250	Programmstart und Tabelleninitialisierung
260 - 290	Umrechnung Hexa in Dezimal
300 - 350	Umrechnung Dezimal in Hexa
360 - 510	DATA-Statements mit den 6502-Mnemonics
520 - 550	Eingabe der Startadresse
560 - 620	Abfrage auf Druckerausgabe
630 - 670	Disassembler-Schleife
680 - 970	BRANCH-Befehle
980 - 1040	Eingabe (Y)es, (N)o, e(X)it
1050 - 1090	RTS-Befehl mit Stack-Abfrage
1090 - 1170	JSR-Befehl mit Stack-Erhöhung
1180 - 1200	JMP-Befehl absolut
1210 - 1250	JMP-Befehl indirekt
1260 - 1300	Ende

Die wichtigsten
Programmteile vom
»Logic Disassembler«

Ende der Disassemblierung gesprungen.

Das Programm »Logic-Disassembler« in der vorliegenden Form für den VC 20 geschrieben, es ist jedoch nach Änderung der Zeilen 100 und 530 (die POKE-Befehle ändern lediglich die Farben des Bildschirms) sofort auf dem Commodore 64 und den anderen CBM-Computern lauffähig.

(Fred Hammer)

SYNTHETISCHE

Mit den »synthetischen Steuerzeichen« stehen Ihnen Möglichkeiten offen, die Sie bisher

Nach dem Einschalten des VC 20 oder des C 64 und der »READY«-Meldung des Computers kann man sich nach Herzenslust auf dem Bildschirm austoben — zunächst ohne Programm. Sie können vorab Texte gestalten, Grafiken entwerfen, dabei die Farben wechseln, hier Zeichen ergänzen und da Stellen abändern. Komfortabel wird dieses Austoben der Möglichkeiten auf der »elektronischen Tafel« jedoch erst durch die Cursor-Steuertasten CLR/HOME, INST/DEL, RVS ON/OFF sowie durch die Farbwahltasten. Man erreicht damit ein hohes Maß an Flexibilität im Ansteuern jeder beliebigen Bildschirmstelle.

Schreibt man nun unversehens ein Anführungszeichen (engl.: quote) auf den Schirm, so verhält sich der Computer plötzlich anders: die Cursor-Tasten wirken nicht mehr — man kommt mit ihnen nicht mehr aus der Zeile heraus. Auch die Farbumschaltung mißlingt, das Bildschirmlöschen mit CLR/HOME versagt, lediglich DEL funktioniert noch. Und dies alles in voller Absicht! Warum? Statt der Ausführung der Steuerbefehle zieht es der Computer vor, sich die Anweisungen in Form reverser Steuerzeichen zu »merken«. Wozu? Nun, um Steuerbefehle dieser Art programmierbar zu machen. (Man stelle sich nur einmal vor, der Cursor sei nicht per Software steuerbar. Verfolgen wir diesen Gedanken lieber nicht weiter.) Zurück zum Thema: Nach dem ersten Anführungszeichen befindet sich der Computer offenbar in einem anderen Verarbeitungsmodus, genannt Quote-Modus, in dem er (mit Ausnahme von DEL) alle Steuerbefehle als Reverszeichen »speichert« und damit zur späteren Ausführung bereitstellt. Der Quote-Modus wird verlassen, sobald das zweite Anführungszeichen einge-

tippt oder die RETURN-Taste betätigt wird. Da die Steuerzeichen — eingeschachtelt in Gänsefüßchen — vom Computer wie Texte behandelt werden, können sie in Programmen auch als solche verarbeitet werden. Dazu stehen alle Stringoperationen zur Verfügung, die auch bei »normalen« Texten verwendet werden. Wirkung zeigen Steuerzeichen jedoch erst dann, wenn man sie mittels PRINT aktiviert. Fazit: Ein Basic-Listing kann normalerweise keine reversen Zeichen enthalten — es sei denn, es handelt sich um Steuerzeichen in Strings.

Steuerbefehle ohne »Gänsefüßchen«

Wir haben uns als Commodore-Anwender sicherlich schon längst an das reverse Q für CURSOR UP oder an das Herzchen für CLR/HOME gewöhnt. Doch es geht auch anders. Das muß es auch, wenn der Drucker zum Beispiel keine Steuerzeichen ausgeben kann. In diesem Fall bedient man sich der CHR\$-Funktion, die durch die Befehlsfolge PRINT CHR\$(X) aktiviert wird. Mit X = 147 wird beispielsweise ebenfalls der Bildschirm gelöscht. Wie kommt das? Tippen Sie bitte ein: PRINT ASC("([CLR/HOME])")

Bitte tippen Sie die Ausdrücke, die in den geschweiften Klammern stehen, nicht als Buchstabenfolge ein, sondern zum Beispiel für PRINT ASC("([CLR/HOME])") nach dem ersten Gänsefüßchen die Control-Taste und dann gleichzeitig die CLR/HOME-Taste. Verfahren Sie bei den folgenden Beispielen dementsprechend.

Nach erfolgtem RETURN lesen Sie: 147. Das reverse Herzchen wird vom Compu-

ter also als Zeichen interpretiert, das den ASCII- beziehungsweise CHR\$-Code 147 trägt. Ein weiteres Beispiel: Die Zeilen PRINT "/RV ON]TEST" und PRINT CHR\$(18)"TEST" bewirken dasselbe, weil PRINT ASC("([CTRL RVS ON])") zeigt, daß der CHR\$-Code des reversen R eben 18 ist.

Ausgehend von der Tatsache, daß jedem Steuerzeichen ein bestimmter Code in der CHR\$-Liste (siehe Handbuch) zugeordnet ist, wurde die Idee geboren, daß diese Zuordnung — um es mathematisch auszudrücken — umkehrbar eindeutig sein müsse. Das heißt, zu jedem Steuerbefehl müßte auch ein entsprechendes reverses Steuerzeichen gehören, das denselben Zweck erfüllt.

Nun gibt es zum Beispiel die Möglichkeit, über PRINT CHR\$(14) auf Kleinbuchstaben umzuschalten. Gibt es dafür auch ein Steuerzeichen? Auf konventionellem Weg hieße die Antwort klipp und klar: nein. Aber ...

Zu Beginn war alles nur Spielerei, bis sich die erstaunlichen Möglichkeiten und Anwendungen häuften. Deshalb: spielen Sie nun bitte mit!

Steuerzeichen auf illegalem Weg

Bevor wir das große Geheimnis lüften, wollen wir erst einmal an ganz und gar legalen Steuerzeichen üben, wie man sie illegal eingibt. Halten Sie sich bitte zunächst streng an das angegebene Rezept, auch wenn es noch andere Eingabeformen gibt. Das gilt auch für die späteren Beispiele. Unsere erste Übung soll darin bestehen, den Bildschirm ohne CHR\$(147) und ohne die Taste CLR/HOME zu löschen — im Direktmodus, versteht

sich. Dazu geben Sie bitte hintereinander in einer Zeile ein:

1. Schritt:

PRINT

2. Schritt:

Erstes Anführungszeichen setzen. Jetzt befinden wir uns im Quote-Modus, den wir sofort wieder verlassen wollen. Daher:

3. Schritt:

Zweites Anführungszeichen setzen.

4. Schritt:

Mit DEL eine Stelle zurückgehen. Dabei wird das zweite Anführungszeichen gelöscht. Da wir uns nicht mehr im Quote-Modus befinden, reagiert der Computer auf Steuerbefehle.

5. Schritt:

Mit CTRL RVS ON auf Revers-Modus schalten. Das ist notwendig, da wir ja ein reverses Steuerzeichen quasi künstlich erzeugen wollen!

6. Schritt:

SHIFT S eingeben. Beim geSHIFTeten S bekommen wir eines der S-Taste zugeordneten Grafikzeichen, in diesem Fall das gewünschte Herzchen. Nun verlassen wir den Revers-Modus:

7. Schritt:

CTRL RVS OFF. Wenn Sie wollen, können Sie noch ein Anführungszeichen zur optischen Abrundung anhängen.

Was jetzt auf dem Bildschirm steht, sieht so aus, als hätten wir nie den legalen Pfad verlassen, als hätten wir immer mit der CLR/HOME-Taste gearbeitet. Dieses Steuerzeichen ist jedoch künstlich entstanden — wird es auch wirken wie ein »echtes«? Das Drücken der RETURN-Taste überzeugt uns schnell davon, daß der Computer uns diese umständliche Manipulation nicht übel genommen hat: es funktioniert tatsächlich!

Bedenken Sie bei der Eingabe, daß nur zwei Anführungszeichen auf dem Schirm erscheinen, obwohl insgesamt drei eingegeben wurden. Der Computer be-

STEUERZEICHEN

nicht für möglich gehalten haben. Ein neuartiger Listschutz ist dabei nur ein Teilaspekt.

findet sich also nach dem dritten Gänsefüßchen wieder im Quote-Modus.

An der Stelle, an der soeben mit SHIFT S das Steuerzeichen für CLR/HOME generiert wurde, kann nun jedes beliebige reverse Zeichen erzeugt werden. Diejenigen davon, die auch eine Steuerfunktion ausüben, aber nicht auf normalem Wege über direkte Tastendrucke erzeugt werden können, wollen wir im folgenden »synthetische Steuerzeichen« nennen.

Die Synthetischen kommen

Wenn es — und das war die Frage, die alles ins Rollen brachte — anstelle der Umschaltung auf Kleinbuchstaben mit CHR\$(14) ein äquivalentes Steuerzeichen gibt, dann muß es zwischem dem reversen E für WHT (Code 5) und dem reversen Q für CURSOR DOWN (Code 17) zu suchen sein. Numeriert man das Alphabet entsprechend durch, so findet man den Buchstaben N unter der Codezahl 14. Das reverse N müßte demnach das Steuerzeichen für die Umschaltung auf Kleinbuchstaben sein. Probieren wir es aus:

```
? " " DEL RVS ON
```

```
N RVS OFF RETURN
```

Es klappt! Der Computer schluckt diesen ungewöhnlichen Befehl und führt ihn aus.

Dieser Erfolg motivierte eine Suche nach allen verfügbaren synthetischen Steuerzeichen; die Ausbeute fiel jedoch zunächst recht mager aus. Die nachfolgende Tabelle zeigt Ihnen die synthetischen Zeichen und ihre Wirkungen bei Bildschirmausgaben (beim Drucker sieht's etwas anders aus — dazu aber später mehr).

Steuerzeichen	CHR\$-Code	Wirkung
H	6	Die manuelle Umschaltung zwischen Groß- und Kleinbuchstaben mit Hilfe der Commodore- und SHIFT-Tasten wird gesperrt.
I	9	Die obige Sperre wird wieder gelöst.
M	13	RETURN. Alles, was nach dem reversen M in der Zeile steht, wird nicht mehr gedruckt. Leider ist eine echte Simulation der RETURN-Taste zum Ändern von Basic-Zeilen unter Programmkontrolle nicht möglich.
N	14	Umschalten auf Kleinbuchstaben.
T	20	DEL. Hiermit können Sie Teile oder komplette Programmzeilen optisch verschwinden lassen. Dieses Steuerzeichen wirkt auch beim Listen eines Programms auf dem Schirm. Anwendung: partieller Listschutz.
SHIFT M	141	SHIFT RETURN. Das Super-Zeichen! Mehr dazu im Text.
SHIFT N	142	Umschalten auf Großbuchstaben.
SHIFT T	148	INS. Kann zum Ändern von Text an bereits gedruckten Text benutzt werden. Beispiel: ? "[SHIFT T]X" liefert ein X auf dem Schirm. bei mehrfacher Abarbeitung wird stets ein weiteres X an die bereits vorhandenen angehängt.

Es sei nochmals darauf hingewiesen, daß die Zeichen dieser Tabelle als reverse Zeichen nach obigem Verfahren erzeugt werden müssen (Ausnahmen: T für DEL und SHIFT T für INS können auch direkt erzeugt werden; sie sind im eigentlichen Sinn keine synthetischen Steuerzeichen.)

Auch wenn die Eingabe der synthetischen Steuerzeichen umständlich erscheint, man wird sich schnell daran gewöhnt haben und auf ihre Vorteile nicht mehr verzichten wollen — nicht nur, weil sie gegenüber der CHR\$-Funktion Speicherplatz sparen.

So unscheinbar und unwichtig das SHIFT RETURN ist, so gewaltig sind die Möglichkeiten, die uns das zuge-

hörige synthetische Steuerzeichen eröffnet. Es handelt sich um das reverse, geschiftete M (Code 141).

Das Super-Steuerzeichen

Gibt man ? "[DEL](TEST [CTRL RVS ON] [SHIFT M] [CTRL RVS OFF] LAUF" ein, so zeigt sich nach dem Drücken der RETURN-Taste, daß der Computer nach dem Teilstring »TEST« einen Wagenrücklauf (carriage return) mit Zeilenvorschub (line feed) durchführt und »LAUF« direkt darunter ausgibt. Genau dieses Verhalten erwartet man von SHIFT RETURN. [SHIFT M] kann jedoch mehr — viel mehr! Das

nächste Beispiel soll es beweisen:

```
10 ? "[DEL] TEST [CTRL RVS ON] [SHIFT M] R [CTRL RVS OFF] LAUF"
```

Geben Sie bitte dieses einzeilige Programm ein und starten Sie es. Wie erwartet erscheint »TEST« und darunter revers »LAUF«. Nun listen Sie bitte dieses Programm. Erstaunt? Das ist tatsächlich neu! Das Steuerzeichen für die Umschaltung in den Revers-Modus erscheint nicht im Listing, sondern wird entgegen aller bisherigen Kenntnisse ausgeführt. »LAUF« wird im Listing revers ausgegeben. Durch das SHIFT-RETURN-Zeichen

wird der Basic-Interpreter offenbar veranlaßt, nachgestellte Steuerzeichen auch im Listing wirksam werden zu lassen. Es zeigt sich, daß diese Schlüsselfunktion alle Steuerbefehle, seien es Farbumschaltungen oder Cursor-Bewegungen, aktiviert. Ungeahnte Möglichkeiten eröffnen sich nun zur optischen Aufbereitung, das heißt Strukturierung und Gestaltung von Listings auf dem Bildschirm. Einige Beispiele sollen dies verdeutlichen.

Beispiel 1:

Reverse REM-Zeile ohne sichtbares REM-Statement und ohne Zeilennummer

```
10 REM " "[DEL] [CTRL RVS ON] [SHIFT M] [SHIFT Q] R [CTRL RVS OFF] TESTLAUF
20 FOR I=1 TO 10
30 PRINT "TESTLAUF":NEXT
```

Die Zeilen 20 und 30 sind Beiwerk, damit die REM-Zeile nicht so allein dasteht. Listen Sie das Programm bitte. Sie sehen die reverse Überschrift »TESTLAUF« ohne REM und ohne Zeilennummer und darunter wie gewohnt die Programmzeilen 20 und 30. Wie funktioniert dieser Trick? Das reverse [SHIFT M] bewirkt einen Sprung zum Anfang der nächsten Zeile und gibt die nachfolgenden Steuerzeichen zur Ausführung frei. Mit dem reversen [SHIFT Q]

geht der Cursor eine Zeile nach oben, also auf »1« der Zeilennummer 10. Reverse R schaltet auf reverse Zeichen um, in denen dann der REM-Text »TESTLAUF« ausgegeben wird. Dabei überschreibt der Computer die Zeilennummer sowie das REM-Statement. Das alles vollzieht sich so schnell, daß es sich der Beobachtung entzieht. Der Anwender sieht nur noch die reverse Überschrift und kann demzufolge an dieser Zeile auch nichts mehr ändern.

Sollten Sie die Original-REM-Zeile noch auf dem Bildschirm haben, so hängen Sie an das »F« von »TESTLAUF« noch das Steuerzeichen für CURSOR DOWN an. Sie werden sehen, daß beim Listen sogar eine Leerzeile zwischen Überschrift und Programm entsteht. Selbstverständlich können derartige REM-Zeilen beliebig im Programm verstreut sein. Wirkung: Im Programmlauf werden unter der reversen Überschrift »Quadratzahlen« die Ergebnisse ausgegeben, während beim Listen des Programms der PRINT-Befehl verschwiegen wird und man eine

REM-Zeile wie in Beispiel 1 vermutet. Damit trägt eine PRINT-Zeile zur optischen Strukturierung eines Listings bei — ein Effekt, der bislang in dieser Form unmöglich schien.

Beispiel 2. Farbige REM-Zeile (ändern Sie nur Zeile 10).

Beispiel 3. Wirkung: Grüne Überschrift ohne Zeilennummer und ohne REM, gelbes Listing.

Beispiel 4. Wirkung: Die bekannten Herzchen sorgen nach dem Listen jeder Zeile für ein Löschen des Bildschirms, so daß die beiden Programmzeilen weder einzeln noch insgesamt sichtbar bleiben und die Codezahlen A und B dem Anwender vorzuenthalten werden.

Mit diesen Beispielen sollen Sie, die Leser vom 64'er, nun an Ihre Computer entlassen werden. Sie werden sicherlich Spaß an der Vielseitigkeit dieser neuen Methode gewinnen und eine Vielzahl von Anwendungen austüfeln.

In der nächsten Folge erfahren Sie dann alles über synthetische Steuerzeichen für den Drucker.

(Jürgen Wagner)

Beispiel 2:

```
10 REM " " [DEL] [CTRL RVS ON] [SHIFT M] [SHIFT Q] [1] [CTRL
RVS OFF] TESTLAUF [CTRL RVS ON] [SHIFT #] [CTRL RVS OFF]
```

Beispiel 3:

Listschutz

```
10 A=4711 : REM " " [DEL] [CTRL RVS ON] [SHIFT M] [SHIFT S]
20 B=0815 : REM " " [DEL] [CTRL RVS ON] [SHIFT M] [SHIFT S]
reverses SHIFT M für SHIFT RETURN
reverses SHIFT S für CLR/HOME
```

Beispiel 4:

REM-Ersatz

```
10 PRINT " " [DEL] [CTRL RVS ON] [SHIFT M] [SHIFT S]
R [CTRL RVS OFF] QUADRATZAHLEN
reverses SHIFT M für SHIFT RETURN
reverses SHIFT S für CLR/HOME
reverses R für RVS ON
20 FOR I=1 TO 10
30 PRINT I,I*I : NEXT
```

Einige Beispiele für die synthetischen Steuerzeichen

Tips & Tricks

Reset für den VC 20

Wem ist das noch nicht passiert? Ein Maschinenprogramm, ein paar POKES und der Computer hat sich aufgehängt.

War das Programm vorher nicht abgespeichert, so ist man verraten und verkauft, da der VC 20 keine eingebaute Reset-Taste hat. Mit einem Minimum an handwerklichem Geschick kann man hier jedoch leicht Abhilfe schaffen. Auf Seite 152 des VC 20-Handbuches ist der User-Port beschrieben. Uns interessieren Pin 1 (Ground) und Pin 3 (Reset). Wenn man diese beiden Pins kurzschließt, wird ein Kaltstart ausgeführt, genau wie bei SYS 64802. Da der Strom nicht abgeschaltet wurde, sind die Speicherinhalte erhalten geblieben. Basic-Programme scheinen zwar verloren gegangen zu sein, können aber durch »Verbiegen« der entsprechenden Zeiger wieder gerettet werden.

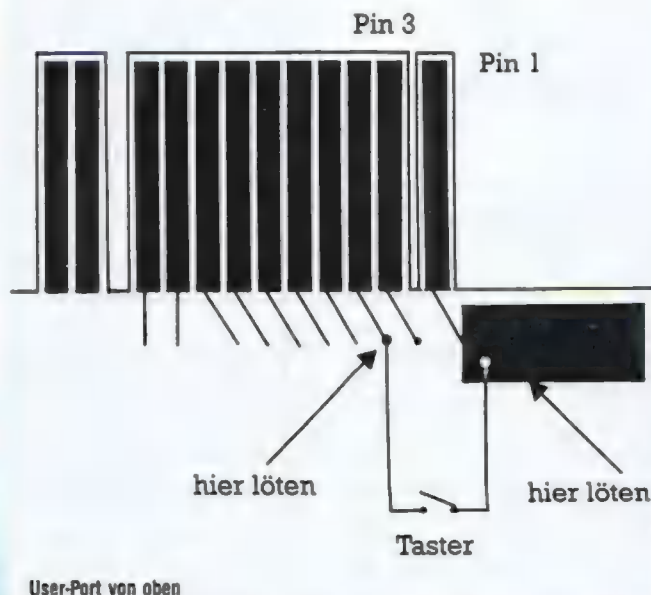
Die eleganteste Lösung eines Kurzschlusses von Pin 1 und Pin 3 ist der Einbau eines Tasters. Wenn

Sie den VC 20 öffnen (bitte nicht während der Garantiezeit), sieht der User-Port wie im Bild dargestellt aus.

Löten Sie zwei isolierte Drähte an den eingegebenen Punkten an und verbinden Sie sie mit dem Taster. Den Taster können Sie im Gehäuse einbauen, zum Beispiel rechts oberhalb der Tastatur, oder Sie legen die Drähte nach außen und lassen den Taster aus dem User-Port hängen.

Wenn Sie Ihren Computer nicht öffnen möchten oder sich die Lötarbeiten nicht zutrauen, können Sie sich einen Stecker für den User-Port besorgen und dort die beiden Drähte zum Taster befestigen. Nach diesem kleinen Eingriff genügt ein kurzer Tastendruck, und der VC 20/C 64 ist wieder betriebsbereit.

(Jörg Grohne)



Der Trick mit dem Autostart

Offt ist es beim Laden von Maschinenprogrammen lästig, sich die verschiedenen SYS-Adressen zum Start zu merken oder sie in dem Berg von Notizen ausfindig zu machen. Da empfiehlt sich ein Starter, der mit »LOAD"name",8,1« geladen wird, sich selbst startet, das Maschinenprogramm auf die richtige Adresse lädt und ebenfalls startet.

```
10 REM *** MARTIN HEINZ UND ANDREAS VON LEPEL ***
100 AD=704
110 READ QP$
120 GOSUB 500
130 POKE AD,QP
140 AD=AD+1
150 IF X<>71 THEN GOTO 110
160 END
500 QL$=LEFT$(QP$,1):QR$=RIGHT$(QP$,1)
510 QL=VAL(QL$):IF QL=0 AND QL$<>"0" THEN QL=ASC(QL$)-55
520 QR=VAL(QR$):IF QR=0 AND QR$<>"0" THEN QR=ASC(QR$)-55
530 QP=QL*16+QR:RETURN
1000 DATA A9,06,85,B7,A9,01,85,B9
1010 DATA 85,B9,A9,08,85,BA,A9,F8
1020 DATA 85,B8,A9,02,85,BC,A9,F8
1030 DATA 8D,02,03,A9,CF,8D,03,03
1040 DATA A9,00,A8,20,D5,FF,4C,F8
1050 DATA CF,00,00,00,00,00,00,00
1060 DATA 00,00,00,00,00,00,00,00
1070 DATA 48,45,53,4D,4F,4E,00,00
1080 DATA 88,E3,C0,02,7C,A5,1A,A7
```

Der Basic-Lader

Dazu muß man wissen, daß das Betriebssystem nach Ausführung der Loadroutine (\$FFD5) indirekt in den Basic-Warmstart springt (\$0302/\$0303). Also wird dieser Vektor auf die Startadresse des Ladeprogramms gesetzt. Um auf der Diskette nur einen Block für den Lader zu belegen, wählt man für ihn den Bereich, der für Sprite 11 reserviert ist (\$02C0-\$02FE).

Dieses einfache Maschinenprogramm erfüllt die geforderten Bedingungen als Lader für das Programm »HESMON« mit Start \$Cff8. Der Anwender muß für ein anderes Hauptprogramm nur die Länge des Namens ändern (\$02C1); den Zeiger auf den Filenamen (\$02Cf/\$02D2), den Namen und die Adresse des Hauptprogramms (\$02D7/\$02DC & \$02E7/\$02E8). Dann wird der Bereich von \$02C0 bis \$0303 vom Monitor aus abgespeichert: »S"LOADER"08 02C0 0303«.

Zum Test des Programms vollziehe man nun einen Kaltstart, und nach »LOAD"LOADER",8,1« erscheint SEARCHING FOR LOADER, LOADING, SEARCHING FOR HESMON, LOADING, dann erfolgt der Start des Programms »HESMON«.

Der Teil des oben genannten Laders, der als Vorbereitung der Loadroutine die Zeropage beschreibt, kann auch durch die Routinen \$FFBA und \$FFBD ersetzt werden.

Der Basic-Lader POKEd das Maschinenprogramm direkt in die zugehörigen Speicherzellen. Achtung: Nach Ablauf des Basic-Laders startet das Maschinencode-Programm direkt. Das Programm wandelt die Hex-DATAs in Dezimalzahlen um und POKEd diese. Das Programm, das geladen werden soll, bestimmt man durch Umändern der jeweiligen Speicherzellen, wie oben beschrieben.

(Martin Heinz und Andreas von Lepel)

Monitor-Auszug:

```
m02c0 0303
:02c0 a9 06 85 b7 a9 01 85 b8
:02c8 85 b9 a9 08 85 ba a9 f8
:02d0 85 bb a9 02 85 bc a9 f8
:02d8 8d 02 03 a9 cf 8d 03 03
:02e0 a9 00 a8 20 d5 ff 4c f8
:02e8 cf 00 00 00 00 00 00 00
:02f0 00 00 00 00 00 00 00
:02f8 48 45 53 4d 4f 4e 00 00
:0300 8b e3 c0 02 7c a5 1a a7
!! !!
```

Basic-Warmstart Vektor

Disassemblierter Lader:

```
02c0 lda #$06
02c2 sta $b7
02c4 lda #$01 ;Laenge des Namens
02c6 sta $b8
02c8 sta $b9 ;Filenummer
02ca lda #$08 ;Secundaeradresse
02cc sta $ba
02ce lda #$f8 ;Geraetennummer
02d0 sta $bb ;LOB
02d2 lda #$02 ;Zeiger auf
02d4 sta $bc ;HOB
02d6 lda #$f8 ; Filenamen
02d8 sta $0302
02db lda #$cf ;Start des Hauptprogr.
02dd sta $0303
02e0 lda #$00 ; in Warmstart-Vektor
02e2 tax ; ! sonst Verify
02e3 jsr $ffd5 ;Loadroutine
02e6 jmp $cfc8 ;Start des Hauptprogrammes
```

Der Monitorauszug und der disassemblierte Lader für den Autostart von Hesmon

Tips und Tricks für den VC 20

Nützliche ROM- und RAM-Adressen für den VC 20 wurden schon mehrfach veröffentlicht. Dieser Artikel bringt nun Beispiele, wie diese auch sinnvoll eingesetzt werden können. Denn was nützt das beste ROM-Adressenlisting mit den tollsten Erläuterungen, wenn man keinen Anwendungszweck findet?

Beginnen wir mit der Ton-erzeugung. Mancher VC 20-Besitzer hat das möglicherweise schon so oft gelesen, daß er in der Nacht von den Adressen träumt. Tongenerator 1 bis 3: 36874 bis 36876, Generator für weißes Rauschen: 36877, Lautstärke: 36878. Doch die wenigsten wissen, daß man mit der Lautstärke auch den Klang eines Tones entscheidend beeinflussen kann, indem man ihn langsam auf- oder abschwellen läßt.

In der Praxis sieht das so aus, daß man einen bestimmten Ton wählt, dann die Lautstärke langsam aufdreht, eine Weile hält und schließlich wieder langsam abdrehet. Wenn man den Ton sofort voll aufdreht und gleich danach langsam abschwellen läßt, klingt es, als ob man auf einem Instrument einen Ton anschlägt, der dann langsam ausklingt. Listing 1 zeigt ein Beispielprogramm mit einer bekannten Melodie.

Die bekannteste Nutzungsmöglichkeit der Lautstärke dürfte wohl die Explosion sein. Man wählt ein tiefes Rauschen und läßt es langsam leiser werden. Das so entstehende Explosionsgeräusch kann sehr gut für Spiele verwendet werden. Das kleine Programm in Listing 2 erzeugt zum Beispiel bei jedem Tastendruck ein anderes Explosionsgeräusch.

Farbige Anleitung

Da wir gerade von Spielen sprechen: Zu einem Spiel gehört auch eine Spielanleitung (auf dem Bildschirm). Und wenn man da einmal et-

was ganz besonderes haben will, gibt es die Möglichkeit, innerhalb eines Buchstaben mehrere Farben zu verwenden. Wie geht das? In der Speicheradresse 646 steht die momentan benutzte Farbe. Zulässig sind Werte von 0 bis 7. Addiert man zu einem dieser Werte 8, so wird auf Vierfarbenmodus umgeschaltet und es entstehen bunte, verzerrte Buchstaben.

Diese lassen sich zwar schlechter lesen, ergeben aber zum Beispiel bei Spielprogrammen einen interessanten Effekt (für Textverarbeitung sind sie weniger geeignet).

Save-Schutz

Die Buchstaben setzen sich aus vier Farben zusammen: Aus Hintergrundfarbe, Rahmenfarbe, Hilfsfarbe und der gewählten Zeichenfarbe aus der Adresse 646. Die Hilfsfarbe wird mit den Bits 4 bis 7 aus 36878 definiert. Es ist unbedingt notwendig, daß die Hintergrundfarbe nur einmal dabei vorkommt. Ist nämlich eine dieser Farben gleich der Hintergrundfarbe, so entstehen Lücken in den Buchstaben, und der Text wird so stark verstümmelt, daß er sich nicht mehr lesen läßt. Das Rücksetzen erfolgt einfach durch Umschalten der Zeichenfarbe mit der CTRL-Taste.

Doch nun zu einem anderen Thema: Im VC 20 gibt es einen Vektor, der jedesmal bei der Ausführung des SAVE-Befehls benutzt wird. Er steht in 818/819 und zeigt auf die SAVE-Routine des VC

20. Durch Ändern dieses Vektors kann man die Ausführung des SAVE-Befehls und damit eventuell das unerlaubte Kopieren eines Programms, verhindern. Zum Beispiel kann dieser Vektor durch POKE 818,116:POKE 819,196 auf \$C474 gesetzt werden. Nach Eingabe des SAVE-Befehls erscheint dann immer direkt READY, ohne daß etwas passiert. Eine andere Möglichkeit ist, eine Fehlermeldung auszugeben. Dies geschieht durch POKE 818,53:POKE 819,196, wodurch der Zeiger auf \$C435 gesetzt wird. Nach der Eingabe von SAVE erscheint »OUT OF MEMORY ERROR«.

Doch die wirksamste Möglichkeit wird wohl die sein, das Programm durch den SAVE-Befehl zu löschen, indem man den SAVE-Vektor auf \$FD22 setzt. Dies geschieht durch POKE 818,34:POKE 819,253. Nach Eingabe des SAVE-Befehls erscheint die Einschaltmeldung, und das Programm ist gelöscht.

Leider ist es mit diesen zwei POKE-Befehlen nicht getan. Denn durch gleichzeitiges Drücken der STOP- und RESTORE-Taste werden sie sofort wieder rückgängig gemacht. Es muß also zusätzlich noch die RESTORE-Taste abgestellt werden. Das geht einfach mit POKE 37150,2. Danach hat die RESTORE-Taste keine Funktion mehr. Wenn nötig, kann sie durch POKE 37150,130 wieder eingeschaltet werden. Dieser POKE-Befehl schreibt in das Interrupt-Enable-Register des VIA 6522 # 1 des VC 20, der für den NMI zuständig ist.

Doch nun wollen wir keine Tasten außer Betrieb setzen, sondern welche abfragen. Oft steht man vor dem Problem, mehrere Funktionen gleichzeitig zu steuern. Leider kann man in Basic immer nur eine Taste auf einmal abfragen, weitere gedrückte Tasten werden normalerweise nicht erkannt.

Zehn Tasten gleichzeitig abfragen

Aber glücklicherweise gibt es beim VC 20 eine Möglichkeit, sieben Tasten auf einmal abzufragen. Die Bits 1 bis 7 der Speicherstelle 145 entsprechen folgenden Tasten: Der linken Shift-Taste, X, V, N, #, ? und Cursor up/down. Die Taste ist gedrückt, wenn das entsprechende Bit gelöscht ist.

Nun läßt sich das ganze mit der Speicherzelle 653 noch erweitern. Bit 0 entspricht den Shift-Tasten, Bit 1 der Commodore-Taste und Bit 2 der CTRL-Taste. Im Gegensatz zu den sieben anderen Tasten sind hier jedoch die entsprechenden Bits bei Tastendruck gesetzt.

Der Computer unterscheidet übrigens zwischen der rechten und der linken Shift-Taste. Mit der Abfrage PEEK (145) AND 2 OR PEEK (653) AND 1 kann man zwischen beiden Shift-Tasten unterscheiden. Sie ergibt normalerweise 2, beim Drücken der linken Shift-Taste 1 und beim Drücken der rechten Shift-Taste 3. Shift lock entspricht der linken Shift-Taste.

(Thomas Gruber/ev)


```

1 REM TON-DEMO
2 REM =====
3 REM
4 REM
5 DIM T(5)
10 FOR X=1 TO 5:READ T(X):NEXT
15 READ T,D:POKE 36875,T(T):POKE 36876,T(T)
20 IF D=-1 THEN END
25 FOR X=15 TO 0 STEP -1
30 POKE 36878,X
35 FOR Y=1 TO 3*D:NEXT Y
40 NEXT X
45 GOTO 15
50 REM
55 REM

```

```

60 DATA 195,201,207,209,215
65 DATA 5,1,3,1,3,6,4,1,2,1,2,6
70 DATA 1,1,2,1,3,1,4,1,5,1,5,6
75 DATA 5,1,3,1,3,6,4,1,2,1,2,6
80 DATA 1,1,3,1,5,1,5,1,1,6,0,1
85 DATA 2,1,2,1,2,1,2,1,2,1,3,1,4,6
90 DATA 3,1,3,1,3,1,3,1,3,1,4,1,5,6
95 DATA 5,1,3,1,3,6,4,1,2,1,2,6
100 DATA 1,1,3,1,5,1,5,1,1,6
105 DATA 0,-1

```

READY.

Listing 1. Ton-Demo

```

1 REM EXPLOSIONEN
2 REM =====
3 REM
4 REM
10 G=36877:L=36878:X=RND(1)*100+130
15 FOR I=15 TO 0 STEP -1
20 POKE G,X+I:POKE L,I
25 FOR D=1 TO 240-X:NEXT D
30 NEXT I
35 POKE G,0:POKE L,0
40 GET A$:IF A$="" THEN 40
45 RUN

```

READY.

Listing 2. Explosionen

Veränderungen am 64-Basic

Eine der merkwürdigsten Effekte am Commodore 64 ist, wenn man unmittelbar nach dem Einschalten den Speicher abfragt, also eingibt `PRINT FRE(0)`

Man erhält nämlich eine negative Zahl als Antwort. Man kann dies korrigieren, indem man bei einer negativen Antwort den Wert 65536 addiert. Im folgenden soll diese Merkwürdigkeit etwas näher beleuchtet und ein Weg zur Korrektur dargestellt werden.

Die Funktion `FRE(0)` liefert den noch verfügbaren, also freien Speicher. Dazu wird der Speicher aufgeräumt (Garbage-Collection), das heißt, die nicht mehr benötigten Strings werden entfernt und der gesamte String-Bereich folglich geordnet. Der freie Bereich ergibt sich dann als: Untere Grenze der Strings minus obere Grenze der Arrays.

Der Inhalt der folgenden Adressen muß also voneinander abgezogen werden

33/34 Untergrenze Strings

31/32 Ende Arrays

also

Inhalt von 33 minus Inhalt von 31

Ergebnis nach 63

Inhalt von 34 minus Inhalt von 32

Ergebnis nach 62.

Das Ergebnis ist also vom Typ `INTEGER`. Allerdings arbeitet der Interpreter intern nicht mit Integer-Zahlen, sondern mit `REAL`-Zahlen (Gleitkommazahlen), das Ergebnis muß also gewandelt werden. Bei der Wandlung wird ein Bit (von 16, die für die Darstellung der Integer benötigt werden) als Vorzeichen betrachtet. Ist der Inhalt von Zelle 62 größer oder gleich 80 = 128, so wird die Zahl bei der Wandlung als negative Zahl angesehen. Der Inhalt von 62 wird dann größer oder gleich 80 sein, wenn der freie Bereich größer 32767 Byte ist.

Aus diesen Gründen taucht gelegentlich eine negative Zahl als freier Speicherplatz auf. Wie kann man dies ändern?

Im Interpreter gibt es vier Wandlungsroutinen.

1) Wandle `REAL`-Zahl nach `INTEGER`.

Dabei wird die Integer-Zahl mit 15 Bit + 1 Bit für das Vorzeichen verschlüsselt, darstellbar sind also als Integer ganze Zahlen von -32767 bis +32767.

2) Wandle `INTEGER`-Zahl nach `REAL`

Dies ist die Umkehrung von 1).

3) Wandle `REAL` nach `INTEGER`.

Dabei wird eine Integer-Zahl mit 16 Bit (kein Vorzeichen!) verschlüsselt, darstellbar sind also Zahlen von 0 bis 65535. Diese Routine wird für die interne Darstellung der Zeilennummern des Basic-Programms benötigt (Wandlung von ASCII nach `REAL`, von dort nach `INTEGER`).

4) Wandle `INTEGER` nach `REAL`.

Dies ist die Umkehrung von 3). Diese Routine wird zum Beispiel für die Ausgabe der Fehlermeldungen benötigt (...ERROR in (Zeile)).

Bei der `FRE`-Funktion müßte also nach der Berechnung des freien Platzes die Routine 4) aufgerufen werden. Aufgerufen wird aber die Routine 2). Im Interpreter steht

`JMP $BC44` (an der Adresse `$B39B`).

Dort müßte aber eigentlich stehen

`SEC`

`JMP $BC49`

Die Frage ist jetzt nur noch, wie kann man dies ersetzen?

Dabei tauchen folgende Probleme auf

1) Der Interpreter wird im ROM gespeichert.

2) Man muß 3 Byte durch 4 Byte ersetzen.

Im Commodore 64 kann man bestimmte Bereiche, die durch ROM belegt sind, ausblenden und dafür RAM einblenden. Das folgende Programm kopiert den Interpreter vom ROM ins RAM

```
10 FOR I=10*16*256 TO 12*16*256-1
```

```
20 POKE I,PEEK(I)
```

```
30 NEXT I
```

Dabei wird ausgenutzt, daß der Computer zwar über `PEEK(I)` aus dem ROM liest, aber mit dem Befehl `POKE` ins RAM schreibt (dies ist intern gelöst).

Man müßte jetzt nur noch das RAM einblenden, vorher sollen aber noch die notwendigen Veränderungen vorgenommen werden.

So einfach kann man natürlich nicht drei Byte durch vier Byte ersetzen. Die Lösung besteht darin, die Routine in einen freien Bereich umzuleiten und dort die notwendigen Befehle einzufügen (sogenannte Patches). Also anstelle von `JMP $BC44` an der Adresse `$B39B`

sollte stehen

`JMP $C000` in den freien Bereich.

Dies erreicht man durch

```
40 POKE 11*16*256+3*256+9*16+12,0
```

(ersetzt \$44 = 44*16+4)

```
50 POKE 11*16*256+3*256+9*16+13,12*16
```

(ersetzt \$BC = 11*16+12).

An der Stelle `$C000` ff. sollte dann stehen

`SEC`

`JMP $BC49`

Also wird wieder `gePOKET`

```
60 POKE 12*16*256,3*16+8 (=SEC)
```

```
70 POKE 12*16*256+1,4*16+12 (=JMP)
```

```
80 POKE 12*16*256+2,4*16+9 ($49 Teil der Adresse)
```

```
90 POKE 12*16*256+3,11*16+12 ($BC Teil der Adresse).
```

Jetzt muß man nur noch das RAM einschalten

```
100 POKE 1,PEEK(1) AND 254 (Einschalten)
```

```
( POKE 1,PEEK(1) OR 1 (Ausschalten) )
```

Im Speicherplatz 1 wird im Bit 0 (= Wertigkeit 1) hinterlegt, wo die Speicherplätze von `$A000` bis `$BFFF` sind.

Und nun, lassen Sie Ihren Computer doch mal den Befehl `PRINT FRE(0)`

ausführen. Wenn Sie keinen Fehler gemacht haben, wird er Ihnen diese Frage ab sofort korrekt beantworten.

Zusammenfassend kann man sagen, daß der beschriebene Effekt unschön und der Aufwand zur Änderung gering gewesen wäre. Der hier beschriebene Aufwand wurde dadurch groß, daß die Änderung nachträglich vorgenommen werden mußte.

(Dr. August Quint)



Underground, so der Originaltitel, ist nicht irgend ein Spiel, sondern eine Kombination aus Action- und Abenteuerspiel.

Für den VC 20 mit mindestens 16 KByte Erweiterung geschrieben ist Underground mit einigen überraschenden Eigenschaften ausgestattet.

In den Untergrund mit

2000 Mark für den Untergrund

»Underground« Gesticht durch die Verknüpfung von Action- und Abenteuerspiel. Wie es zu den Abenteuern im Untergrund von New York kam, schildert der Autor selbst.

Ich hatte letztes Jahr zwischen den Abiprüfungen irgendwann mal eine Woche Zeit, um etwas Längeres zu programmieren. Da ich zu der Zeit gerade eine gute Labyrinth-Routine in Maschinensprache geschrieben hatte, die mich damals einigermaßen faszinierte, war das Grundthema eines Spiels eigentlich schon vorgegeben.

Es sollte ein mit Joystick steuerbares, nicht aggressives Spiel werden, bei dem Action und Überlegung in etwa eine gleiche Rolle spielen sollten. Zu dem Zweck habe ich mir erst einmal ein Männchen definiert, das mittels Joystick einigermaßen realistisch bewegt werden konnte. Beim Testen des Bewegungsablaufes dieses Männchens kam dann die Idee, den Ablauf des Spiels in den Untergrund einer Stadt zu legen, und so entstand dann der kleine Vorspann.

Das Zahlenlabyrinth schwirrte mir eigentlich schon seit einiger Zeit im Kopf herum, und die Programmierung gestaltete sich relativ einfach. Als ich mir überlegte, noch ein wenig Action in das Spiel hineinzubringen, entstanden die beiden Aufzugsszenen. Das Schlußbild ergab sich aus der Überlegung, ein einfach zu berechnendes, aber trotzdem schwer zu spielendes Bild zu schaffen, bei dem ich auch die Labyrinthroutine noch ein wenig benutzen konnte.

Aufgrund dieser Szenen entstand dann das eigentliche Spiel UNDERGROUND. (Oliver Joppich)

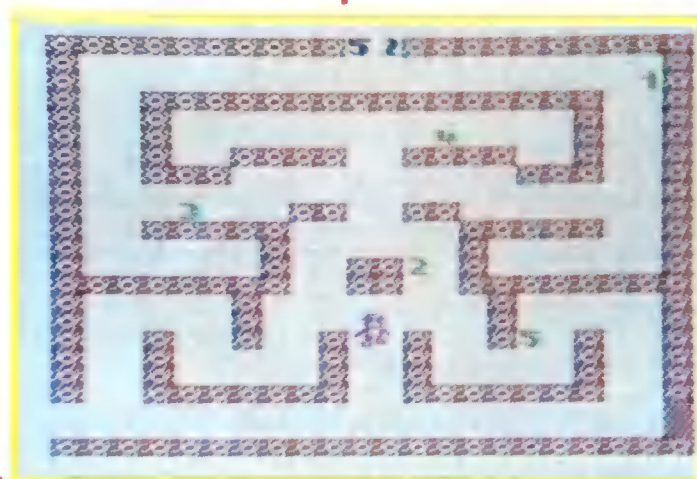


Oliver Joppich

Das Ziel des Spiels ist es, in möglichst kurzer Zeit aus dem Untergrundsystem von New York zu entkommen und dabei alle gestellten Aufgaben richtig zu lösen.

Das Spiel besteht aus insgesamt fünf verschiedenen Spielszenen und einer kleinen Vorgeschichte, in deren Verlauf man durch einen offenen Schacht in das unterirdische System von New York stürzt. Durch Drücken der Feuertaste am Joystick ist es jetzt möglich, in die erste Spielszene zu gelangen.

▼ Bild 3. Das Labyrinth der wilden Zahlen

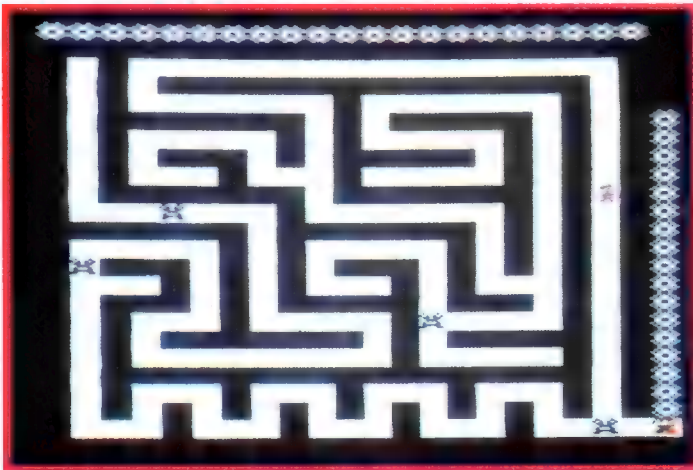




Listing des Monats

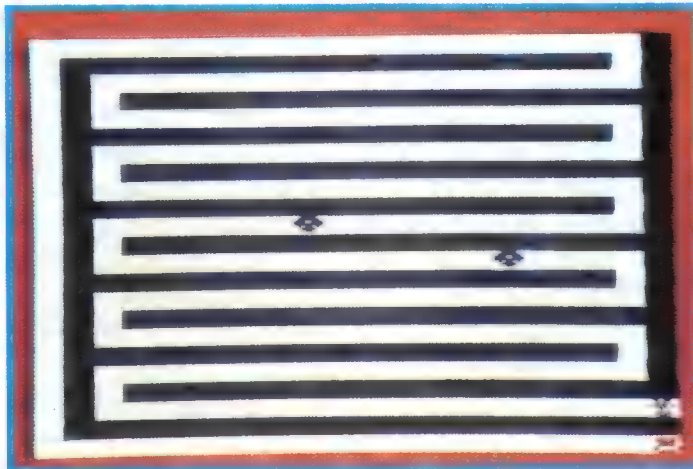
dem *VE* 20

Bild 1 (oben) und Bild 2 (rechts). Nach dem Sturz in den Untergrund steht man vor einer Paternosterkaskade



▲ Bild 4. »Piplips« einsammeln — in einem sich ständig verändernden Labyrinth

▼ Bild 5. Dies ist das letzte Hindernis, das es zu überwinden gilt



Benutze Maschinenspracheroutinen (Aufruf mit SYS) ▼

- 4864 Joystickabfrage; nach Aufruf dieser Routine kann mit PRINT PEEK(1001) der Schußknopf und mit PRINT PEEK(1000) die Richtung abgefragt werden.
- 7360 Malt die Aufzüge auf den Bildschirm.
- 5000 Crattatazong!! Soundroutine 1.
- 7296 Hyaehh!! Soundroutine 2.
- 7168 Verschieben der Aufzüge um eine Einheit nach oben.
- 7224 Labbymaker, die wohl interessanteste Maschinenspracheroutine. Mit ihr wird innerhalb einer Zehntelsekunde ein Labyrinth auf den Bildschirm gemalt.

x,y	die Koordinaten des Männchens
lo	die Geschwindigkeit des Männchens
sc	Screenstart (=4096)
co	Start des Farb-RAM (=37888)
po	Position des Männchens am Bildschirm
fa	Position des Männchens im Farb-RAM
jo	Adresse der Joystickroutine (=4864)
a(1)...a(5)	Position der 5 magischen Zahlen
zq	Bildschirmpoke für Piplip
pi	Bildschirmpoke für Piplip
we	Joystickrichtung
le	Level
ze	Anzahl der mitgenommenen Zahlen

Wichtige Variable

10 - 135	Initialisierung
136 - 190	Titelbild
198 - 260	Männchen einen Schritt zur Seite
266 - 290	Titelbild New York, Schrift
298 - 840	Vorspannszene
846 - 990	Aufzüge Bild 1
996 - 1590	Das Zahlenlabyrinth
1596 - 1790	Piplips fangen
1796 - 1950	Aufzüge Bild 2
1956 - 2110	Die Glibberbahn
2120 - 2440	Ende
2450 - 2530	da fehlt eine Nummer oder ein Piplip
2580 - 2690	damit geht's auf und ab (das Männchen)
2700 - 2710	Level := Level + 1
2720 - 2730	Timeout bei Piplipcatch
2740 - 2750	Ton langsam aus
2760 - 2760	just for fun

Die wichtigsten
Unteroutinen


```

5 IFPEEK(44)<32THENPRINT"BITTE ERST EINGEBEN";POKE44,32:PG192,0:"NEW":END
10 CLR:FOR=0TO315:READA:POKE4864+T,A:SU=SU+A:NEXT
20 IFSUC>222400THENPRINT"FEHLER IN DATEN":END
30 PRINT"ABSPEICHERN"

```

Listing 2. Maschinenspracheroutinen und Grafik für »Underground«

[illegible]

In dieser ersten Szene (Bild 2) geht es darum, ohne Berührung der Wände von der linken auf die rechte Seite zu gelangen. Um dies zu erreichen, gilt es, die Aufzüge, die sich von unten nach oben bewegen, geschickt zu benutzen.

Die wilden Zahlen

Nach diesem ersten, noch recht einfachen Bild gelangt man in das Labyrinth der wilden Zahlen (Bild 3). Hier geht es darum, in möglichst kurzer Zeit (maximal 1 Minute) die Zahlen des Labyrinthes in ihrer Reihenfolge aufzusammeln. Damit diese Arbeit ein wenig erschwert wird, tauschen die Zahlen alle sieben Schritte ihre Position in zufälliger Folge. Um diese kleine Gemeinheit aber ein wenig auszugleichen, kann auch der Spieler die Position der Zahlen verändern, indem er den Schußknopf betätigt. Auch dann tauschen zwei zufällig ausgewählte Zahlen ihre Position auf dem Spielfeld. Wenn alle Zahlen aufgesammelt sind, muß man sich zum Ausgang auf der rechten unteren Seite begeben.

Man kann dieses Bild auch verlassen, wenn man nicht alle Zahlen aufgesammelt hat. Dadurch ist es möglich, sich auch die folgenden Spielszenen anzuschauen. Allerdings gelangt man zum Schluß des Spiels leider nicht wieder an die Oberfläche der Stadt, sondern muß nochmal von vorn anfangen.

Die Piplips

In der nächsten Szene (Bild 4) besteht die Aufgabe darin, alle Piplips aufzusammeln, das sind die kleinen schwarzen Wesen, die überall im Bild verteilt sind und danach zum Ausgang rechts unten zu gelangen. Die Schwierigkeit dieses Bildes besteht darin, daß man nur eine bestimmte Anzahl von Schritten zur Verfügung hat, um von einem Pipip zum anderen zu gelangen. Jedesmal, wenn man eines der Wesen aufgesammelt hat, erscheint sofort ein neuer Irrgarten mit anders


```

32086 DATA0.16,0.152,160,208,180,189,0.2,16,190,153,253,1,198,123,169,255,133
32087 DATA122,96,165,43,166,44,160,1,133,95,134,96,177,95,240,31,200,200,165
32088 DATA21,209,95,144,24,240,3,136,208,9,165,20,136,209,95,144,12,240,10,136
32089 DATA177,95,170,136,177,95,176,215,24,96,208,253,169,0,168,145,43,200,145
32090 DATA43,165,43,24,105,2,132,45,165,44,105,0,133,46,32,142,166,169,0,208
32091 DATA45,32,231,255,165,55,164,55,133,51,132,52,165,45,164,46,133,47,132
32092 DATA48,133,49,132,50,32,29,168,162,25,134,22,104,168,104,162,250,154,72
32093 DATA152,72,169,0,133,62,133,16,96,24,165,43,105,255,133,122,165,44,105
32094 DATA255,133,123,96,144,6,240,4,201,171,208,233,32,107,189,32,19,166,32
32095 DATA121,0,240,12,201,171,208,142,32,115,0,32,107,169,209,134,104,104,165
32096 DATA20,5,21,208,6,169,255,133,20,133,21,160,1,132,15,177,95,240,67,32,44
32097 DATA168,32,215,170,200,177,95,170,200,177,95,197,21,208,0,0,0,0,0,0,0,0
32098 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
32099 DATA24,133,0,160,0,177,254,153,252,17,177,0,145,254,200,177,254,153,252
32100 DATA17,177,0,145,254,200,200,200,192,20,208,230,165,1,133,255,24,165,0
32101 DATA133,254,105,22,133,0,169,0,101,1,133,1,201,18,240,21,160,0,177,0,145
32102 DATA254,200,177,0,145,254,200,200,200,192,20,208,240,76,42,28,130,152,10
32103 DATA169,15,141,14,144,134,0,198,0,208,252,169,0,141,14,144,134,0,198,0
32104 DATA208,252,202,208,201,89,96,96,96,96,96,96,96,96,96,96,96,96,96,96,96,96
32105 DATA141,13,144,141,132,144,141,11,144,141,10,144,141,14,144,162,255,160
32106 DATA255,141,14,144,202,208,250,169,255,77,14,144,141,14,144,202,208,253
32107 DATA135,152,41,240,170,192,64,208,233,96,96,96,96,96,96,96,96,96,96,96,96
32108 DATA96,96,96,96,96,160,0,169,128,153,0,16,153,0,17,169,4,153,0,148,153
32109 DATA0,149,200,208,237,76,80,19,96,96,96,96,96,96,96,96,96,96,96,96,96,96
32110 DATA133,0,160,0,153,134,148,200,209,250,96,96,96,96,96,96,96,96,96,96,96,96
32111 DATA96,96,96,96,0,169,160,160,0,153,22,16,153,0,17,153,0,148,153,0,149
32112 DATA200,208,241,32,160,29,234,234,169,45,133,254,169,16,133,255,32,209
32113 DATA29,24,133,38,24,41,3,170,32,0,30,160,0,177,2,201,160,208,32,24,165
32114 DATA2,133,0,165,3,133,1,138,145,2,32,80,30,169,32,160,0,145,2,165,0,133
32115 DATA254,165,1,133,255,76,32,29,232,138,24,41,3,170,165,38,24,41,3,133,38
32116 DATA228,38,240,4,138,76,42,29,160,0,177,254,170,169,32,145,254,160,0,138
32117 DATA224,4,48,1,96,32,0,31,165,3,133,255,165,2,133,254,76,32,29,0,0,0,0
32118 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
32119 DATA0,169,91,145,0,24,165,0,105,22,133,0,169,0,101,1,133,1,201,18,208,233
32120 DATA169,5,141,45,16,159,91,160,0,153,0,16,200,192,22,208,248,96,173,22
32121 DATA145,77,21,145,77,24,145,77,25,145,96,169,100,96,0,0,0,0,0,0,0,0,0,0,0
32122 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
32123 DATA254,133,2,169,0,101,255,133,3,76,73,30,224,3,208,16,24,169,44,101,254
32124 DATA133,2,169,0,101,255,133,3,76,73,30,224,2,208,16,56,165,254,233,2,133
32125 DATA2,165,255,233,0,133,3,76,73,30,56,165,254,233,44,133,2,165,255,233
32126 DATA0,133,3,96,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
32127 DATA255,133,3,76,73,30,224,3,208,16,24,169,22,101,254,133,2,169,0,101
32128 DATA133,3,76,73,30,224,2,208,16,56,165,254,233,1,133,2,165,255,233,0,133
32129 DATA3,76,73,30,56,165,254,233,22,133,2,165,255,233,0,133,3,96,96,0,0,0
32130 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
32131 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
32132 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
32133 DATA16,24,169,2,101,254,133,2,169,0,101,255,133,3,76,73,31,224,1,208,16
32134 DATA24,169,44,101,254,133,2,169,0,101,255,133,3,76,73,31,224,0,208,16,56
32135 DATA165,254,233,2,133,2,165,255,233,0,133,3,76,73,31,56,165,254,233,44
32136 DATA133,2,165,255,233,0,133,3,96,0,0,0,165,254,133

```

```

10 REM                                     U N D E R G R O U N D
20 REM WRITTEN BY
30 REM OLIVER JOPPICH
40 REM CELLER STR.52
50 REM 3300 BS
60 :
70 CLR POKE36879,0:PRINT"U":POKE36869,205:PRINTCHR$(8):CO=37988:SC=4096:PO=SC:L
E=0
80 JO=4864:REM JOYSTICKROUTINE
81 LO=60:POKE7645,96
90 OJ$(0)=" "
100 OJ$(2)=" "
110 OJ$(3)=" "
120 AM$="JKLNMOPQRSTUVWXYZ[ ]_:POKE0,0:REM BRAUCHT JEMAND NEIN KRANKENWAGEN
130 POKE36876,220:POKE36877,254:FORT=0T0500:NEXT:POKE36876,0
135 :
136 REM TITELBILD 1
137 :
140 FORY=1TO19STEP4:
150 FORX=0TO20:POKE646,RND(1)*7+1
160 SYSJO:IFPEEK(100)=1THEN#50
170 GOSUB200:PRINTMID$(OJ$(Y/4),X+1,1):POKE36878,15:POKE36878,0
180 NEXT:PRINT:PRINT" ":NEXT
190 FORT=0TO3000:NEXT:GOTO270
197 :
198 REM BEWEGUNG DES MAENNCHENS
199 :
200 POKEPO,32:POKEPO+1,32:POKEPO+22,32:POKEPO+23,32
210 FA=CO+X*Y*22:A=4:POKEFA,A:POKEFA+1,A:POKEFA+22,A:POKEFA+23,A
220 PO=SC+X*Y*22:POKEPO,2:POKEPO+1,3:POKEPO+22,18:POKEPO+23,19
230 FORT=0TOLO:NEXT
240 PO=SC+X*Y*22:POKEPO,0:POKEPO+1,1:POKEPO+22,16:POKEPO+23,17
250 FORT=0TOLO:NEXT:
260 RETURN
265 :
266 REM TITEL 2
267 :
270 PRINT"U"
275 PRINT" "
280 PRINT" "
290 FORT=0TO9999:NEXT
297 :
298 REM SCENE 1 IN NEW YORK

```

verteilten Piplips. Tip: 1. Erst in Ruhe das Labyrinth anschauen und dann lossausen. 2. Auch dieses Bild kann man ohne alle Piplips verlassen, allerdings gelten auch hier dieselben Einschränkungen wie bei den Zahlen.

Aufzüge und Labyrinth

Als nächstes kommt jetzt wieder ein Bild mit Aufzügen, allerdings um einiges schneller und mit kleineren Fahrstühlen. Auch hier muß man möglichst die Nerven behalten und von links nach rechts zum Ausgang gelangen.

Und nun zur letzten Szene (Bild 5): Auf den ersten Blick vielleicht ein simples Labyrinth mit zwei regelmäßig umherfliegenden Objekten (eine Namensgebung sei diesmal jedem selbst überlassen). Die Aufgabe besteht darin, zum Ausgang zu gelangen, ohne mit den fliegenden Objekten zusammenzustößen. Sollte es aus Versehen doch zu einem Zusammenstoß kommen, wird man leider in das vorletzte Bild zurückgeschleudert und muß erst mal wieder die Aufzüge überwinden.

Die Schwierigkeit bei diesem Bild besteht darin, daß nach einmaligem Losgehen in eine bestimmte Richtung die Bewegung nicht mehr gestoppt werden kann, bis man an einer Wandbegrenzung zum Stillstand kommt. Wer jetzt immer noch glaubt, daß dieses Bild leicht zu schaffen sei, möge bitte das Programm abtippen, zu einem Joystick greifen und anschließend spielen.

Das Eintippen

Um Underground auf möglichst einfache Art und Weise vom Papier in den Computer zu bringen, ist das Programm in drei Teile unterteilt:

Listing 1: Lader.

Dieser dient dazu, den Maschinen- und Basic-Teil von Underground von Kassette/Diskette zu laden. Um dieses Programm benutzen zu können, muß noch der

[illegible]

Teil 2 sowie der Teil 3 eingegeben werden.

Listing 2: Maschinensprache/Grafik.

In diesem Teil befinden sich die Maschinenroutinen und die Grafik des Spiels. Nach dem Eintippen sollte man das Programm zwar aus Sicherheitsgründen abspeichern, aber es wird bei der lauffähigen Version nicht mehr gebraucht, da es den notwendigen Programmteil selbst auf Kassette/Diskette ablegt.

Listing 3: Basic.

Hier ist das Hauptprogramm des Spiels in Basic abgelegt.

Vorgehensweise:

* Lader abtippen und abspeichern

* POKE44,32:POKE8192,0:
NEW eingeben.

* Maschinensprache/Grafik
abtippen und abspeichern

* Maschinensprache/Grafik mit RUN laufen lassen (er-

stellt Maschinenspracheteil des Programms) und danach abspeichern, indem man auf die obere Zeile geht und **<RETURN>** drückt.

* Computer kurz ausschalten und wieder einschalten (wegen der verbogenen Pointer)

* Basic-Teil abtippen und abspeichern mit SAVE"UG. BAS" (.8 bei Floppy).

* Wenn man Besitzer einer Floppy ist, war das alles und man kann das Spiel laden und starten mit »LOAD"LA-DER".8 und RUN).

* Als Besitzer einer Datensette empfiehlt es sich, den Lader, den Maschinenteil und den Basic-Teil in dieser Reihenfolge auf Kassette zu bringen und dann durch Laden und Starten des Laders zu beginnen.

(Oliver Joppich)

```

5 IFLTHEN50
10 PRINT"7 UNDERGROUND LADER
20 PRINT"00(1) TAPE ODER (8) DISK
30 POKE198,0:WAIT198:1:GETA$:A=VAL(A$):IFAC>1ANDAC>8THEN30
35 PRINT"DATEN
40 IFL=0THENL=1:LOAD"UG.DATA",A,1
50 POKE192,0:POKE44,32:POKE828,A:POKE46,74:CLR
55 PRINT"HAUPTPROGRAMM
60 LOAD"UG.BAS",PEEK(828)

READY.

```

Listing 1. Das Ladeprogramm

Listing 1. Das Ladeprogramm


```

1150 PRINT"+      +      +      +";
1160 PRINT"+      +      +      +";
1170 PRINT"+ + + + + + + +";
1180 PRINT"+ + + + + + + +";
1190 PRINT"+ + + + + + + +";
1200 PRINT"+ ++++++ ++++++ +";
1210 PRINT"          <";
1220 PRINT"          <";
1230 PRINT"+++++"; POKE4601,91:POKE38393,2
1240 X=0:Y=20:GOSUB210:LO=15:A$=TI$:TI$="000000"
1250 ZE=1:FORT=1705
1260 A=RND(1)*70+(T-1)*90+30:IFPEEK(A)<>32THEN1260
1270 R(T)=A:POKEA,112+T:POKEA+33792,RND(1)*5+2:NEXT
1280 POKE37998,0:POKE37999,0
1290 POKE4107,VAL(RIGHT$(TI$,1))+112:POKE4106,VAL(MID$(TI$,5,1))+112
1300 IF TI>3600THEN2730
1310 SYSJO:JW=PEEK(1000):IFPEEK(1001)=1THEN1340
1320 IF JW=23THEN1290
1330 Z9=-NOTZ5AND15:IFZ5<>0THEN1390
1340 POKE36878,15:POKE36876,244:POKE36877,0:POKE36875,0:POKE36874,0
1350 Z1=INT(RND(1)*5+1):IFPEEK(R(Z1))<1120RPEEK(R(Z1))>121THEN1380
1360 Z2=INT(RND(1)*5+1):IFZ2=Z10RPEEK(R(Z2))<1120RPEEK(R(Z2))>131THEN1380
1370 Z3=PEEK(R(Z1)):POKEA(Z1),PEEK(R(Z2)):POKEA(Z2),Z3
1380 POKE36878,0:POKE36876,0
1390 XX=0:YY=0:P2=0:SP=30+X+Y*22+23-PEEK(1000):IFJW=22THENP1=1:P2=23:XX=1:JU=1
1400 IFJW=24THENP1=0:P2=22:XX=-1:JU=2
1410 IFJW=1THENP1=22:P2=23:YY=1:JU=3
1420 IFJW=45THENP1=0:P2=1:YY=-1:JU=4
1430 IFP2=0THEN1290
1440 P3=PEEK(SP+P1):P4=PEEK(SP+P2):IFP3=910RP4=911THEN1290
1450 QW=11:IFP3>910RP4>911THENGOSUB1530
1460 IFQW=0THEN1290
1470 X=XX+X:Y=YY+Y:IFX=20THENTI$=A$:GOTO1580
1480 IFX=20THENX=X-1:GOTO1290
1490 IFY<1THENY=1:GOTO1290
1500 ONJUGOSUB200:710,2580,2640
1510 POKE36878,15:POKE36876,0:GOTO1290
1520 GOTO1290
1530 QW=0
1540 IFP3=ZE+1120RP4=ZE+1122THENZE=ZE+1:QW=1
1550 IFQW=1THENPOKE36878,15:POKE36876,220:FORT=19070240:POKE36878,T:NEXT:POKE36876,0:POKE36878,0
1560 IFZE=9THENQW=2
1570 RETURN
1580 POKE36877,234:FORT=0T0158STEP,2:POKE36878,T:POKE36871,T/2:POKE36849,T/2:NEXT
1590 FORT=0T01000:NEXT:GOSUB200:FORT=0T022:POKE4603,T,150:NEXT:PO=4140
1595
1596 REM CATCH THE PIPLIPS !!!
1597
1600 LE=1:GOSUB2700:ZQ=6
1610 POKE36878,0:POKE36876,0:POKE36879,0:SYS7424:POKE36879,24:POKE36877,0
1611 ZQ=ZQ-1
1620 CP=4117:IFZQ<1THEN1660
1630 FORT=1T020
1640 PI=30+RND(1)*505:IFPEEK(PI)<>32THEN1640
1650 POKEPI,103:POKEPI+33792,0:NEXT
1660 POKEPO,106:POKEPO+33792,4:POKE36878,15:POKE4601,102:POKE38393,2:POKE4600,32
1670 SYSJO:WE=PEEK(1000):IFWE=23THEN1670
1680 CC=CC+1AND3:IFCC=0THENPOKECP,160:CP=CP+22:POKE36877,240:POKE36878,15
1690 IFCP>4579THEN1770
1700 IFCC=1THENPOKE36877,0
1710 POKEPO,32:PO=PO+23-WE:IFPEEK(PO)<>32ANDPEEK(PO)<>103ANDPEEK(PO)<>102THENPO=PO-23+WE
1720 IFPEEK(PO)=103THENPOKE36877,0:POKE36878,15:FORT=0T05:FORT=233T0239
1721 POKE36876,T
1730 IFPEEK(PO)=103THENPOKEPO+33792,RND(1)*3:NEXT:NEXT:GOTO1610
1740 IFPEEK(PO)=102THEN1900
1750 POKEPO,106:POKEPO+33792,4:POKE36878,15:POKE36878,0
1760 FORT=0T0000:GOTO1670
1770 POKE36879,8:PRINT"          LE=6"
1780 POKE36876,220:POKE36877,0:POKE36878,0:FORT=0T0224:POKE36878,T:NEXT
1781 POKE36876,0
1790 FORT=0T02000:NEXT:GOTO1600
1795
1796 REM DER AUFZUEGE ZWEITER TEIL
1797
1800 POKEPO,106:POKEPO+33792,4:SYS5000:FORT=0T0500:NEXT:GOSUB2700
1810 POKE36879,30:POKE4953,2:SYS7360
1820 FORT=2T020STEP2:AA=INT(RND(1)*19):AA=4096+T+AA*22
1830 POKEAA,32:POKEAA+1,32:POKEAA+22,32:POKEAA+23,32:POKEAA+44,22:POKEAA+45,22
1840 IF T/4=INT(T/4)THENPOKEAA+56,32:POKEAA+57,32
1850 NEXT:POKE36877,0:POKE36876,0:POKE36875,0:POKE36874,0:LO=50
1860 Y=19:X=0:PO=932:GOSUB200:POKE7255,110
1870 SYSJO:IFPEEK(1000)<>22THENGOSUB2540:GOTO1870
1880 PO=30+X+Y*22:X=X+2
1890 IFPEEK(PO+2)<>220RPEEK(PO+24)<>32THENX=X-2:GOTO1950
1900 X=X-1:GOSUB200:X=X+1:GOSUB200:POKE36878,15:POKE36879,0:POKE7256,PEEK(7255)-10
1910 IFX=20THEN1950
1920 GOTO1870
1930 PRINT"          POKE36879,8:POKE36877,150:FORT=15T00STEP-,2:POKE36878,T:NEXT
1940 GOTO1810
1950 FORT=0T0200:NEXT:GOSUB2700
1955

```

Listing 3. Das Basicprogramm von »Underground« (Fortsetzung)

Druckfehler-teufelchen



Hier bin ich wieder. Auch in der Ausgabe 5 des 64'er Magazins ist es mir mit viel Mühe gelungen, erfolgreich zuzuschlagen. Mir gelangen die Attacken vor allem dort, wo die Herren Redakteure in Zeitnot gerieten. Dann passierte es, daß sie sich auf die Angaben der Programmeinsender verlassen mußten. Diese sind nämlich durchaus in der Lage, ein Listing zu schicken, das mit der aktuellen, lauffähigen Version nicht übereinstimmt.

Das ist dann ein gefundenes Fressen für mich und ich schlage erbarungslos zu.

Supergrafik, Seite 81:

In Zeile 4030 muß es heißen ... THEN BC=BC+1: ...

Wörter werfen, Seite 87:

In Zeile 260 müssen die 42 mal Space nicht hinter dem Ausführungszeichen stehen, sondern direkt nach dem reversen Herz, innerhalb der Anführungszeichen.

Den größten Erfolg hatte ich im Programm

Diskettenzauberei, Seite 89:

Dort gelangen mir gleich 4 Fehler:

Zeile 1120: Hinter PRINT#1 muß ein Komma stehen.

Zeile 1200 und 1260: Schreiben Sie vor dem Doppelpunkt ein Semikolon.

Zeile 1250: Hinter B-P muß ein Doppelpunkt stehen.

Zeile 1260: Schreiben Sie G% anstatt R%

[illegible]

Listing 3. Das Basicprogramm von »Underground« (Schluß)

Druckfehler- teufelchen

Zeile 1280: Tauschen Sie das Semikolon gegen ein Komma aus.

Buchbesprechung:

Commodore 64. Der Computer für Einsteiger und Aufsteiger. Seite 96:

Kostet nicht 49,80 Mark,
sondern 29.80 Mark.

Commodore 64 — Programmieren leicht gemacht

Seite 97

Auch dieses Buch konnte ich teurer machen als es ist: Anstatt 49 Mark ist es bereits für 34 Mark zu haben.

Grafikkurs, Seite 110, Spalte 1:

Die Überschrift soll heißen: »Die Enttarnung ...« und nicht die Entartung.

Seite 111, Spalte 1, 17. Zeile von unten:

Dort muß natürlich stehen: \$FF und nicht \$F.

Seite 112, Spalte 1, 21. Zeile von oben:

Gemeint ist die Tabelle

Seite 115, Tabelle 2:

In der Spalte mit dem Label-Namen muß es unter FRETOP heißen MEMSIZ statt MEMZIZ

Seite 113:

Hier habe ich gewürfelt
und ganze Absätze
durcheinander gewor-
fen:

Spalte 1, unten: Bis zu dem Satz »Nehmen wir an, ein Byte sähe binär so aus: 1111 1011« ist noch alles in Ordnung. Dann muß man weiterlesen: Spalte 2, Zeilen 12-10 von unten. Danach folgt die in Spalte 1 gezeigte AND-Operation, in die sich beim Ergebnis aber noch ein Fehler geschlichen hat. Es muß nämlich herauskommen: 0000 1011. Weiterlesen sollte man nun in Spalte 2, Zeile 9 von unten bis Spalte 3, Zeile 3 von oben. Erst danach folgt der Text aus Spalte 1, Zeile 3 von unten: »Halt!...« bis Spalte 2, Zeile 13 von unten: »...übersehen (Tabelle 4)«. Schließlich geht es dann normal weiter ab Spalte 3, Zeile 4 von oben.



**Nun ist es endlich
so weit! Wir gehen an die
praktische Entwicklung
von Programmen heran.**

**Außerdem wird
Strubs noch um ein
kleines RENUMBER-
Programm erweitert.
Die Syntaxdiagramme der
Basic-Erweiterungen
werden das Arbeiten mit
Strubs erleichtern.**

Wenn Sie Quellprogramme schreiben, achten Sie vor allem auf folgendes: Jeder der neuen Befehle muß durch ein Ausrufezeichen gekennzeichnet werden, zum Beispiel !REPEAT, und benötigt — außer !DO — eine eigene Programmzeile. Folgende Konstruktion ist also zum Beispiel nicht erlaubt:

```
40 !REPEAT X=X+1 !UNTIL X>5
```

Marken beginnen grundsätzlich mit einem Pfund-Zeichen £.

Um die Korrektheit von Konstruktionen zu überprüfen, können Sie die Syntax-Diagramme in Bild 1 benutzen. Wenn sich ein Weg entlang der Linien finden läßt, der der Konstruktion entspricht, dann ist diese in Ordnung. Sie können sich aber auch an den verschiedenen Beispielen im Rahmen dieser Artikelfolge orientieren. Das beste Beispiel für ein Quellprogramm erhalten Sie, wenn Sie die Programmdiskette mit dem Programm Strubs bestellen

(wird allerdings erst ab der nächsten Ausgabe angeboten). Diese Diskette enthält neben dem lauffähigen Objektprogramm, das in der letzten Ausgabe abgedruckt wurde, auch das ausführlich dokumentierte Quellprogramm von Strubs, dessen Abdruck aus Platzgründen nicht möglich ist.

Bei der Blockschachtelung ist darauf zu achten, daß sich verschiedene Blöcke nicht überschneiden dürfen und daß jeder Block korrekt abgeschlossen wird. Hierbei kann man sich immer das Beispiel der FOR-NEXT-Schleifen in Basic vor Augen halten. Vollkommen unmöglich ist beispielsweise folgende Konstruktion:

```
10 !REPEAT
20 : !WHILE ... !DO
30 !UNTIL ...
40 : !EWHILE
```

Nun wird es aber allmählich Zeit, mit der Praxis zu beginnen. Laden Sie das Programm Strubs in Ihren Computer und starten es mit »RUN«. Nun erscheint ein Menü. Geben Sie hier »E« ein, um in den Editbereich zu gelangen (siehe dazu die erste Folge). Der Computer meldet sich mit »READY«, das heißt also, Sie befinden sich jetzt im Direktmodus. Hier können Sie nun (fast) so arbeiten, als sei Strubs gar nicht vorhanden. Geben Sie zunächst »NEW« ein. Jetzt können Sie das kleine Programm aus Bild 2 eintippen und wie sonst gewohnt mit »SAVE "RENUMBER.QP",8« abspeichern.

Mit diesem »QP« hat es folgende Bewandnis: Bei Compilern ist es allgemein üblich, die verschiedenen Files, die zu den einzelnen Phasen der Übersetzung gehören, einheitlich zu kennzeichnen, der Austro-Compiler, arbeitet zum Beispiel mit den Files »name«, »p/name«, »z/name« und »c/name«. Um Quellprogramme und Objektprogramme auseinanderhalten zu können, sollten Sie sich entsprechend von Anfang an daran gewöhnen, dem Programm immer ein »QP« für Quellprogramm beziehungsweise ein

»OP« für Objektprogramm hinzuzufügen. Nun kann das Programm übersetzt werden. Geben Sie ein »!RETURN« und es erscheint das Menü von Strubs. Die Übersetzung wird mit »U« gewählt. Strubs fragt nun nach dem Namen für das Objektprogramm. Geben Sie ein: RENUMBER.OP. Da das übersetzte Programm direkt auf Diskette geschrieben wird, achten Sie darauf, daß die Floppy eingeschaltet ist. Nun erscheint auf dem Schirm die Meldung »1. Lauf«, gefolgt von der Ausgabe der Blockstruktur. Nach Beendigung des 2. Laufs sollte die Meldung »0 FEHLER« erscheinen. Ist dies der Fall, dann können sie mit »E« wieder in den Edit-Bereich gelangen. Hier steht immer noch das Quellprogramm. Um sich das übersetzte Programm anzusehen, laden Sie es mit »LOAD "RENUMBER.OP",8«. Es sollte mit dem Listing in Bild 2 übereinstimmen. Aber starten sie das Programm nicht.

Jetzt übersetzen Sie einmal genauso verschiedene kleine Testprogramme — zum Beispiel die Beispiele aus der letzten Folge — und sehen sich die Ergebnisse an. Dabei werden Sie feststellen, daß einige Bedingungen im Objektprogramm in negierter Form erscheinen. Daß Basic keine boolschen Variablen kennt, hat eine wichtige Konsequenz: Beim Test, ob eine Variable ungleich 0 ist, darf man nicht — wie dies normalerweise häufig in Basic formuliert wird — beispielsweise schreiben

```
IF A THEN ...
```

sondern muß bei jeder Bedingung die vollständige Form

```
IF A <> 0 THEN ...
```

verwenden. Dies liegt daran, daß die Bedingung NOT(A) außer für —1 immer erfüllt ist.

Zweitens kann man sehen, daß in den Objektprogrammen manchmal neue Zeilen auftauchen, die das Quellprogramm nicht enthielt. Strubs generiert solche Zeilen als Sprungziele. Damit immer Platz für solche Zeilen ist, sollte der Abstand der Zeilennummern im Quellprogramm immer mindestens 2 betragen.

Ist bisher alles wie oben beschrieben verlaufen, dann können Sie sich freuen. Sind irgendwelche Fehler aufgetreten, dann vergleichen Sie noch einmal genau das Testpro-

ein Precompiler für Basicprogramme (Teil 3)

ZE: Zeilenende
 ZNR: Zeilennummer
 TRENNZEICHEN: ',', '\times', '\div', '\cdot', '\cdot', ZE
 U: Blank (Leerzeichen), kann außer innerhalb von Marken- und Befehls-Namen beliebig stehen
 NAME:
 beliebige Zeichenfolge ohne Trennzeichen
 Befehls- und Markennamen müssen durch ein Trennzeichen abgeschlossen werden

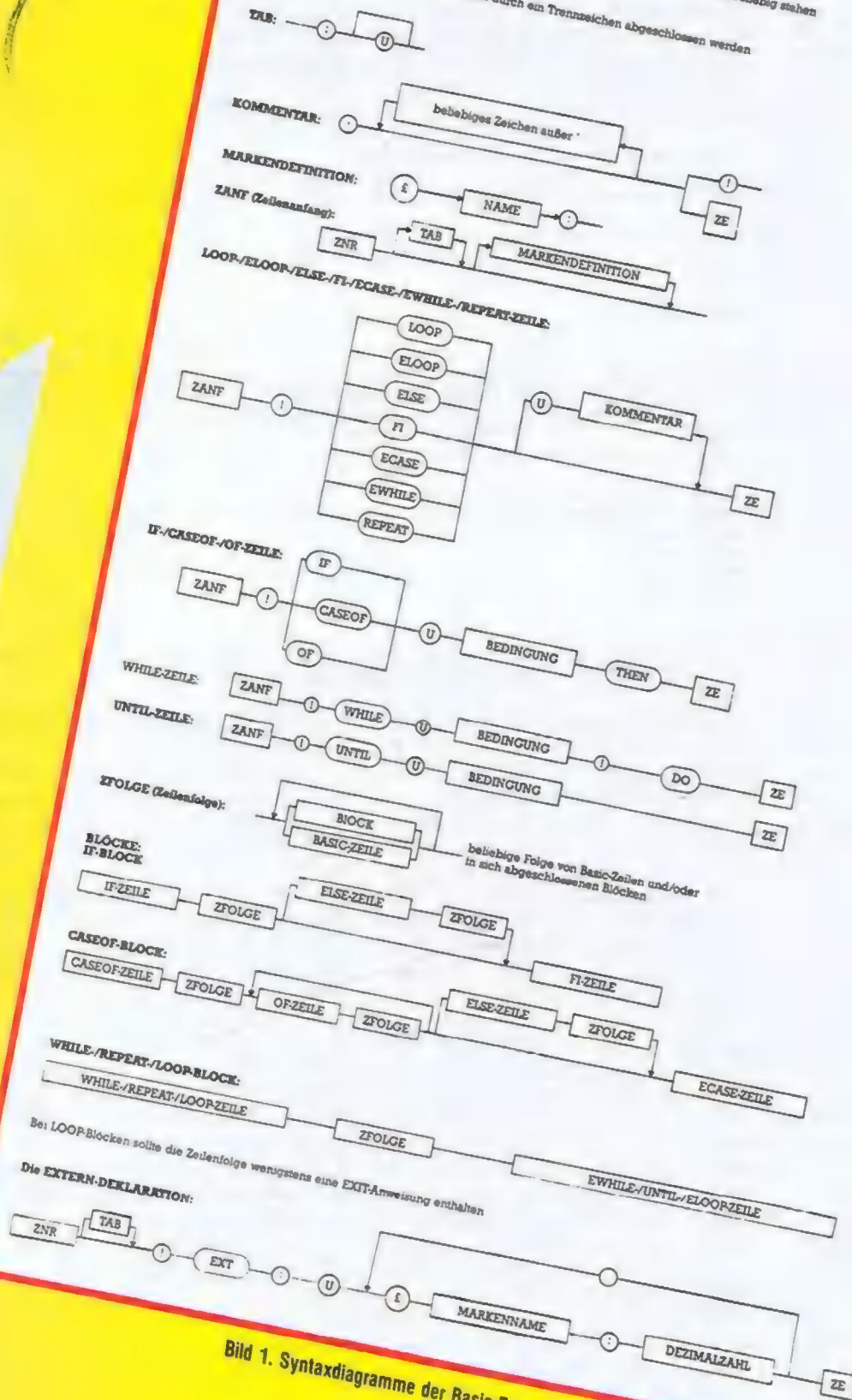


Bild 1. Syntaxdiagramme der Basic-Erweiterungen von Strubs

gramm mit dem Listing in Bild 2 und hoffen Sie, daß der Fehler hier liegt. Wenn Sie keine Abweichungen feststellen, dann haben Sie Pech — Sie haben das Programm Strubs falsch eingetippt.

Wie steht es aber mit Fehlern im Quellprogramm? Syntax-Fehler können sich auf drei verschiedene Arten bemerkbar machen: Vor allem Fehler, die nicht mit den neuen Befehlen zusammenhängen, führen wie gewohnt beim Lauf des Objektprogramms zu den bekannten Fehlermeldungen. Fehler in bezug auf die neuen Befehle quittiert Strubs mit Abbruch der Übersetzung, falls eine Fortsetzung nicht sinnvoll erscheint, oder mit Eintragung in eine Fehlerliste und gleichzeitiger Kennzeichnung der fehlerhaften Zeile im Objektprogramm. Die Fehlerliste kann man sich mit »F« ansehen.

Ein Abbruch der Übersetzung mit entsprechender Fehlermeldung am Bildschirm erfolgt vor allem bei Verstößen gegen die Blockstruktur (und bei Speicherplatzproblemen wie Stack-Overflow oder Listen voll). Bei Fehlern mit den Blöcken — zum Beispiel vor einem ELSE fehlt das IF oder zu einem WHILE fehlt das EW-HILE etc. — gibt es ein Problem, das nicht nur bei Strubs, sondern generell bei allen Compilern auftaucht. Der Fehler wird nicht an der Stelle seines Auftretens bemerkt, sondern erst viel später. Die Zeilennummer bei einer Fehlermeldung wie »BLOCKSCHACHTELUNG: ANFANG FEHLT«, sagt also nichts weiter aus, als daß der Fehler erst hier bemerkt wurde. Um bei der Suche nach dem Fehler zu helfen, gibt Strubs aber während der Übersetzung ein Schema der Blockstruktur aus, mit dessen Hilfe sich solche Fehler leicht lokalisieren lassen.

Bei Meldungen wie »zu viele Marken«, »zu viele WHILE/REPEAT« etc. empfiehlt es sich, das Programm in kleinere Teile zu zerlegen, diese getrennt zu übersetzen und anschließend wieder zusammenzufügen.

Wie man dazu vorgeht, behandeln wir weiter unten. Die entsprechenden Listen sind allerdings so großzügig dimensioniert, daß dieser Fall sehr selten eintreten wird.

Sollte während der Übersetzung aus irgendeinem Grund ein unkontrollierter Programmabbruch erfolgen (zum Beispiel OUT OF MEMORY ERROR), dann empfiehlt es sich mit »GOTO 5000« dafür zu sorgen, daß offene Disk-Files ordnungsgemäß geschlossen werden.

Die Beseitigung von Fehlern, die Strubs bei der Übersetzung entdeckt, gestaltet sich relativ einfach: Notieren Sie sich die Zeilennummern zu jedem Fehler und schalten in den Editbereich (mit »E«). Dort kann das Quellprogramm geändert werden, dann wird mit »!« und Wahl von »U« neu übersetzt. Da das Quellprogramm solange im Edit-Bereich bleibt, bis es durch »NEW« oder Laden eines anderen Programms gelöscht wird, kann dieser Vorgang solange wiederholt werden, bis der

können, erfordert auf der anderen Seite allerdings eine gewisse Disziplin, damit die Verbindung zum Quellprogramm nicht verloren geht. Jede vorgenommene Änderung sollte sorgfältig notiert und nicht zu viele Änderungen auf einmal vorgenommen werden. Dann wird wieder das Programm Strubs und das Quellprogramm (in den Editbereich) geladen. Verbessern Sie das Quellprogramm entsprechend Ihren Notizen und übersetzen es erneut. Dieser Vorgang wird solange

```
54990 ***** RENUMBER *****
55000 RENUMBER: PRINT "J" ***** RENUMBER *****
55010 ZA=40*256+1:INPUT "X0 STARTNR. ";ZN:INPUT " SCHRITTWEITE":SW
55020 !LOOP
55030 IFPEEK(ZA+1)=0 THEN !EXIT 'FERTIG
55040 HZ=ZN/256:POKE ZA+2,ZN-HZ*256:POKE ZA+3,HZ:ZA=PEEK(ZA)+256*PEEK(ZA+1)
55050 ZN=ZN+SW
55060 !LOOP
55070 RETURN

READY.
```

Bild 2. RENUMBER-Quellprogramm für Strubs

```
55000 PRINT "J" ***** RENUMBER *****
55010 ZA=40*256+1:INPUT "X0 STARTNR. ";ZN:INPUT " SCHRITTWEITE":SW
55020 :
55030 IFPEEK(ZA+1)=0 THEN 55061
55040 HZ=ZN/256:POKE ZA+2,ZN-HZ*256:POKE ZA+3,HZ:ZA=PEEK(ZA)+256*PEEK(ZA+1)
55050 ZN=ZN+SW
55060 GOTO 55020
55061 :
55070 RETURN

READY.
```

Bild 3. RENUMBER-Objektprogramm für Strubs

letzte Fehler beseitigt ist. Sobald die Übersetzung mit der Meldung »0 Fehler« beendet wird, geht es ans Testen des Objektprogramms.

Hierzu wird Strubs durch Eingabe von »S« verlassen. Dadurch wird ein Kaltstart ausgeführt, der die Interpretererweiterung abschaltet und den Speicher säubert. Nun laden Sie das Objektprogramm unter dem Namen, den Sie bei der Übersetzung angegeben haben und starten es mit RUN. Dieses Programm wird nun wie jedes normale Basic-Programm ausgetestet. Dazu können selbstverständlich auch Toolkits mit TRACE- und DUMP-Funktionen verwendet werden. Da die Zeilennummern denen des Quellprogramms entsprechen, empfiehlt es sich, ein Listing des Quellprogramms zur Hand zu haben. Fehler in der Programmlogik lassen sich damit leicht finden und beheben.

Die Bequemlichkeit, die Strubs dadurch bietet, daß Programmänderungen und Verbesserungen im Objektprogramm selbst vorgenommen und sofort ausgetestet werden

wiederholt, bis das Ergebnis zufriedenstellend ist.

Dieser soeben beschriebene Ablauf kann allerdings in den meisten Fällen vereinfacht werden: Bis auf zwei Ausnahmen können Objektprogramme auch direkt im Editbereich getestet werden. Damit entfällt die Notwendigkeit, Strubs für jede Übersetzung neu zu laden. Nach der Übersetzung wird mit »E« der Editbereich gewählt, dort das Objektprogramm geladen und getestet. Anschließend wird wieder das Quellprogramm in den Editbereich geladen, verbessert und mit »!« und »U« neu übersetzt und so weiter.

Bei Programmen, die nicht im Editbereich getestet werden können, handelt es sich 1. um Programme, die an einer festgelegten Stelle im Speicher stehen müssen. Strubs selbst ist solch ein Programm. Es muß unbedingt am normalen Basic-Anfang stehen. Solche Programme sind allerdings ziemlich selten. Häufiger dagegen findet sich der 2. Fall: Programme, die den Speicherbereich von 700 bis 800

verändern. Hier steht die in Folge 1 erwähnte Interpreter-Erweiterung von Strubs. Dadurch sind vor allem Programme betroffen, die in diesem Bereich Sprites oder Maschinenprogramme benutzen.

Kommen wir noch einmal auf das Schreiben und Editieren von Quellprogrammen zurück. Bisher haben wir nur davon gesprochen, daß die Programmtexte im Editbereich editiert wurden. Diese Methode hat insbesondere bei der Entwicklung umfangreicher Programme einen

Sie im Programm die Zeilen 45600 bis 45680 einfach weglassen. Damit fallen dann die oben erwähnten Beschränkungen weg, das heißt die unter Fall 2 erwähnten Programme können im Editbereich getestet werden und Strubs kann zusammen mit einer Programmierhilfe benutzt werden. Aber editieren Sie keine Quellprogramme unter Simons Basic. Dazu sind weitere Anpassungen erforderlich, auf die wir in der nächsten Folge näher eingehen. Insbesondere wenn ein Programm aus

sollen. Hierbei ist darauf zu achten, daß sich die Bereiche der Zeilennummern nicht überschneiden. Weisen Sie jedem Programmteil einen bestimmten Zeilennummernbereich zu und verlegen diesen Teil gegebenenfalls mit RENUMBER in diesen Bereich. Anschließend werden nun in jedem Programmteil alle externen Routinen, die dieser Teil aufruft, mit Hilfe der EXTERN-Deklaration vereinbart (das sind die Routinen, die erst nach der Übersetzung angefügt werden). Das oben für die Variablennamen gesagte gilt hier entsprechend. Jetzt können die einzelnen Teile getrennt übersetzt und anschließend in der richtigen Reihenfolge verknüpft werden.

Falls Sie keine Erweiterung besitzen, dann können Sie Strubs um eine RENUMBER-Routine erweitern: Fügen Sie die Zeilen aus Bild 3 in das Objektprogramm von Strubs ein — und, falls Sie das Quellprogramm besitzen, dort entsprechend die Zeilen aus Bild 2. Um nun diese Routine anzubinden, müssen nur noch zwei Zeilen in das Menü eingefügt werden:

```
40110 PRINT "{CDOWN} [REV ON] R
[REV OFF]ENUMBER"
und
40210 IF L$="R" THEN GOSUB
55000: GOTO 40050
beziehungsweise für das Quellprogramm:
40210 IF Z$="R" THEN GOSUB
£RENUMBER: GOTO £MENUE
```

Diese Routine kann dann mit »R« aufgerufen werden, um ein Programm, das sich im Editbereich befindet, umzunummerieren. Das Programm Strubs arbeitet nicht mit der Datasette, sondern es benötigt eine Floppy. Damit der für Strubs unterhalb des Edit-Bereichs reservierte Platz nicht überschritten wird, ist darauf zu achten, daß beim Eintippen des Programms keine Blanks eingefügt werden.

Der Editbereich beginnt bei 40*256+1. Vor dem 1. Start von Strubs läßt sich mit 'PEEK(46) feststellen, ob Strubs diese Grenze nicht überschreitet (der Wert muß kleiner als 40 sein, im Originalprogramm liegt er bei 34).

Da Strubs den Zeiger für »Variablen-Anfang« heraufsetzt, sollte es immer von dem 1. Start abgespeichert werden (auch bei Veränderungen). Sollte man dies einmal vergessen, kann man durch »POKE 46,39:CLR« vor den Abspeichern Strubs in die richtigen Grenzen weisen. (Mathias Törk)

```
40050 PRINT "J"; " *****"
40052 PRINT " * -- STRUBS -- *"
40053 PRINT " * PRECOMPILER *"
40055 PRINT " * BITTE WAEHLEN *"
40058 PRINT " *****"
40060 PRINT "XUEEDIT"

45600 I=0:READW
45610 POKE704+I,W:I=I+1:READW:IFW<256THEN45610
45620 DATA32,115,0,8,201,33,240,4,40,76,231,199
45630 DATA169,18,133,44,169,133,76,231,199,999
```

Bild 4. Diese Änderungen sind an Strubs vorzunehmen, um es auf einen VC 20 mit mindestens 16 KByte Speichererweiterung lauffähig zu machen (zusätzlich zu den Änderungen in Ausgabe 5, Seite 117).

Nachteil: Da Strubs selbst mit einer Interpreter-Erweiterung arbeitet, kann man nicht gleichzeitig andere Interpreter-Erweiterungen — zum Beispiel Toolkits oder das DOS — benutzen. Möchte man auf Befehle wie MERGE, AUTO, FIND etc. nicht verzichten, dann kann man die Quellprogramme vollkommen unabhängig von Strubs entwickeln und erst anschließend das fertige Quellprogramm in den Editbereich laden.

Es zeigt sich, daß die meisten Beschränkungen bei der Arbeit mit Strubs ihren einzigen Grund in der kleinen Interpreter-Erweiterung haben. Wie bereits in der ersten Folge erwähnt, besteht der einzige Sinn dieser Erweiterung darin, das Starten von Strubs vom Editbereich aus dadurch bequemer zu gestalten, daß die Befehlsfolge POKE 44,8: RUN

durch Eingabe von »!« abgekürzt werden kann. Wenn Sie bereit sind, diese Befehlsfolge jedesmal von Hand einzugeben, können Sie auf die Erweiterung verzichten, indem

fertigen Modulen zusammengesetzt werden soll, sind solche Programmierhilfen erforderlich.

Dieser Vorgang verläuft auf der Quellprogramm-Ebene aufgrund der Unabhängigkeit von Zeilennummern relativ einfach. Die einzelnen Programmteile werden in beliebiger Reihenfolge zusammengesetzt. Dazu kann ein Toolkit oder auch das kleine MERGE-Programm aus dem 64'er, Ausgabe 4/84, benutzt werden. Dabei können ruhig gleiche Zeilennummern auftreten und auch die Reihenfolge der Zeilennummern ist beliebig. Anschließend wird der fertige Programmtext mit Hilfe einer RENUMBER-Routine vernünftig durchnummeriert. Da Basic keine lokalen Variablen kennt, ist allerdings vor dem Zusammensetzen auf die Variablennamen zu achten. Um unerwünschte Seiteneffekte zu vermeiden, sind eventuell einige Umbenennungen vorzunehmen. Etwas aufwendiger gestaltet sich der Prozeß, wenn verschiedene Programmteile erst auf der Objekt-Ebene zusammengesetzt werden

Reise durch die Wunderwelt der Grafik

Bevor wir die dritte Etappe auf dem Weg zur hochauflösenden Grafik beginnen, soll noch eine kurze Rückschau auf den in der letzten Folge bewältigten Weg gehalten werden: Wir können jetzt mit dem Binär- und dem Hexadezimalsystem umgehen und haben gelernt, wie man Zahlen darin ausdrückt. Wir haben gesehen, wie der Computer sich Adressen merkt und konnten mit diesem Wissen RAM-Bereiche schützen. Wir haben wichtige Zeiger in unserem Computer kennengelernt. Das Setzen oder Löschen

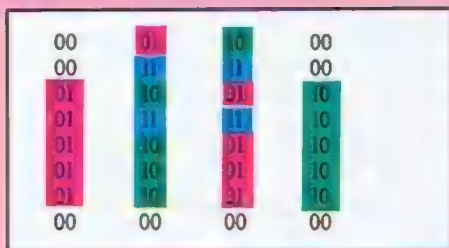


Bild 1. Der Buchstabe A, dargestellt im Mehrfarbenmodus

```

XXX0XXXX (53270), Bit 4 = 0
OR
00010000 Maske, dez. 16
= XXX1XXXX (53270), Bit 4 = 1
  
```

Bild 2. Das Setzen einzelner Datenbits mittels logischer OR-Verknüpfung

von beliebigen Bits in einem Byte beherrschen wir ebenso wie die Definition und Benutzung eigener Zeichen. Wenn Sie seit der letzten Folge viel ausprobiert und so Übung gewonnen haben, dann kennen Sie Ihren Computer schon ganz gut.

Heute werden wir zunächst etwas mehr Farbe ins Spiel bringen, indem wir mehrfarbige Zeichen oder Zeichen mit einem anderen Hintergrund benutzen. Dann betreten wir das Gemach von Dornröschen (der hochauflösenden Grafik) und küssen es wach — im übertragenen Sinne natürlich. Nach dieser Ankündigung sind die ersten schon aufgebrochen. Also gehen wir auch los.

Erinnern Sie sich bitte an Bild 11 aus der letzten oder an Bild 8 aus der ersten Folge. Egal, ob es sich um Zeichen aus dem Zeichen-ROM oder um selbstdefinierte Zeichen handelt, eines haben sie gemeinsam: Überall dort, wo sich im Bit-Muster

In dieser Folge werden wir die ersten Schritte in das Reich der hochauflösenden Grafik wagen. Doch zunächst soll die Grafik etwas farbiger werden. Das heißt, wir erzeugen mehrfarbige Zeichen und variieren die Hintergrundfarbe.

des Zeichens eine 1 befindet, also ein Bit gesetzt ist, taucht bei der Darstellung auf dem Bildschirm ein Punkt in der Zeichenfarbe auf, die im Bildschirmfarbspeicher angegeben ist. Diese Farbe haben wir entweder im Direktmodus durch Drücken der »CTRL«- oder »COM-MODORE«-Taste in Kombination mit einer Zifferntaste festgelegt oder

chen einzeln interpretiert. Es gibt nun auch andere Möglichkeiten der Interpretation des Bit-Musters, und eine davon wird im Mehrfarbenmodus angewendet. Hier sind die Bits des Zeichens zu Paaren zusammengefaßt (Bild 1).

Das ist der Buchstabe »A« mit paarweise zusammengefaßten Bits. Es ergeben sich vier verschiedene

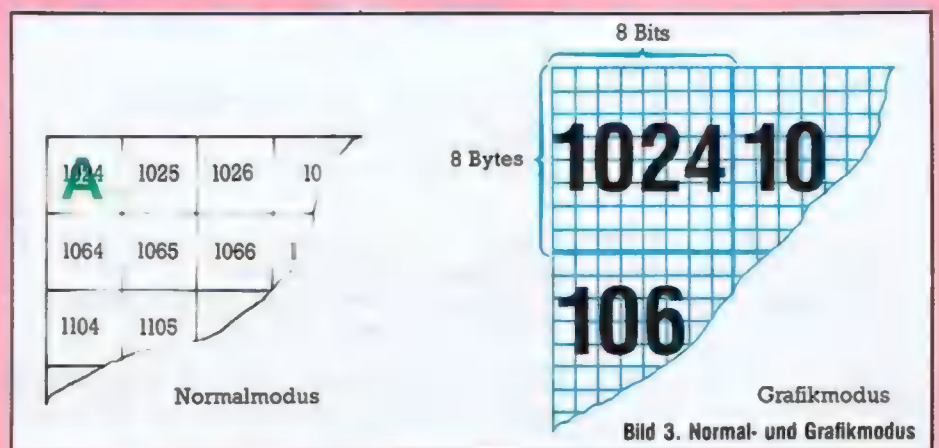


Bild 3. Normal- und Grafikmodus

durch ein »POKE 646, Zeichenfarbe« oder auch durch POKEn eines Farbcodes in die entsprechende Stelle des Bildschirmfarbspeichers.

Möglichkeiten der Bit-Kombination: 00, 01, 10, 11

Erinnern Sie sich an die Binärzahlen, dann erkennen Sie, daß diese

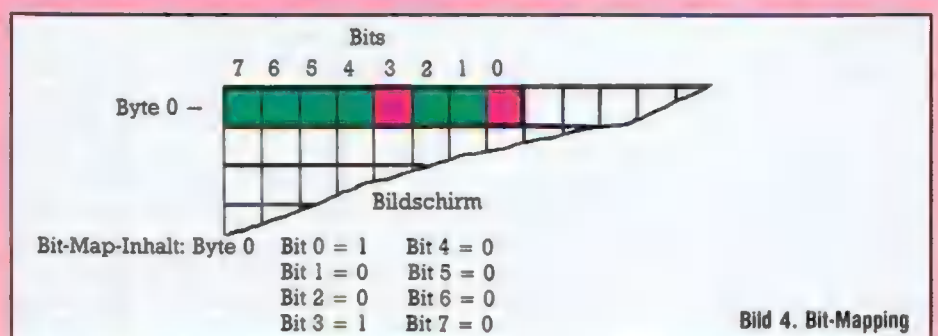


Bild 4. Bit-Mapping

Überall dort, wo im Bitmuster des Zeichens eine 0 ist, taucht bei der Darstellung auf dem Bildschirm nichts auf, das heißt dort erscheint die Farbe des Bildschirmhintergrundes. Jedes Bit aus dem Bit-Muster wird also im Normalbetrieb und auch bei selbstdefinierten Zei-

vier Möglichkeiten den Zahlen 0, 1, 2, 3 entsprechen. Sie wirken wie Zeiger, die in verschiedene Register des VIC-II-Chip deuten (Tabelle 1; siehe auch Registerübersicht in der ersten Folge).

Dort also, wo ein Bitpaar 00 steht, wird es in der Farbe des Bildschirm-

Teil

3

hintergrundes angezeigt, wie bisher im Normalmodus mit einem Bit 0 geschehen. In Adresse 53281 haben wir bisher ja immer schon die Bildschirmfarbe festgelegt. Hinzu kommen die Register 34 und 35 des VIC-II-Chips entsprechend den Adressen 53282 und 53283. Hier können wir nun weitere Farben eingeben, die an den Stellen auf dem Bildschirm gezeigt werden, an denen die dazugehörige Bit-Kombination (01 oder 10) steht. Erlaubt sind alle Farben (0 bis 15).

Nun sind es noch zwei Voraussetzungen, die uns vom Mehrfarben-Zeichen trennen: Dem Computer muß mitgeteilt werden, daß er das Bit-Muster in Paaren statt einzeln interpretieren soll. Das geschieht dadurch, daß man im Register 22 (Adresse 53270) das Bit 4 auf 1 setzt. Falls Sie sich nicht mehr so genau erinnern, wie wir sowas in der letzten Folge gemacht haben — hier ist es nochmal gezeigt (Bild 2).

Es muß also eingegeben werden: POKE 53270, PEEK(53270) OR 16

Dieses Byte 53270 im VIC-II-Chip ist ein Beispiel dafür, wie sorgfältig man auf das richtige Setzen oder Löschen von Bits achten sollte. Wenn Sie sich die Registerübersicht in der 1. Folge und dort besonders dieses Byte ansehen, stellen Sie fest, daß fast jedes Bit eine andere Funktion zu erfüllen hat.

Sollten Sie die obige Basic-Zeile

im Direktmodus eingegeben haben, dann verwandeln Sie alle Buchstaben auf dem Bildschirm in mehrfarbige Zeichen. Auf einem Schwarzweiß-Gerät sind sie dann kaum mehr zu erkennen. Hier stellt man auch den Unterschied zwischen einem Farbfernseher und einem Farbmonitor fest. Der letztere trennt die unterschiedlichen Farben

deutlich voneinander, während auf dem Fernsehschirm häufig Farbmischungen auftreten. Deswegen sollte die Farbgebung in die Register 34 und 35 individuell ausfallen: Probieren, probieren ...

Das Ausschalten des Mehrfarben-Zeichen-Modus geschieht durch Löschen des Bit 4 in 53270 mit POKE 53270, PEEK(53270) AND 239.

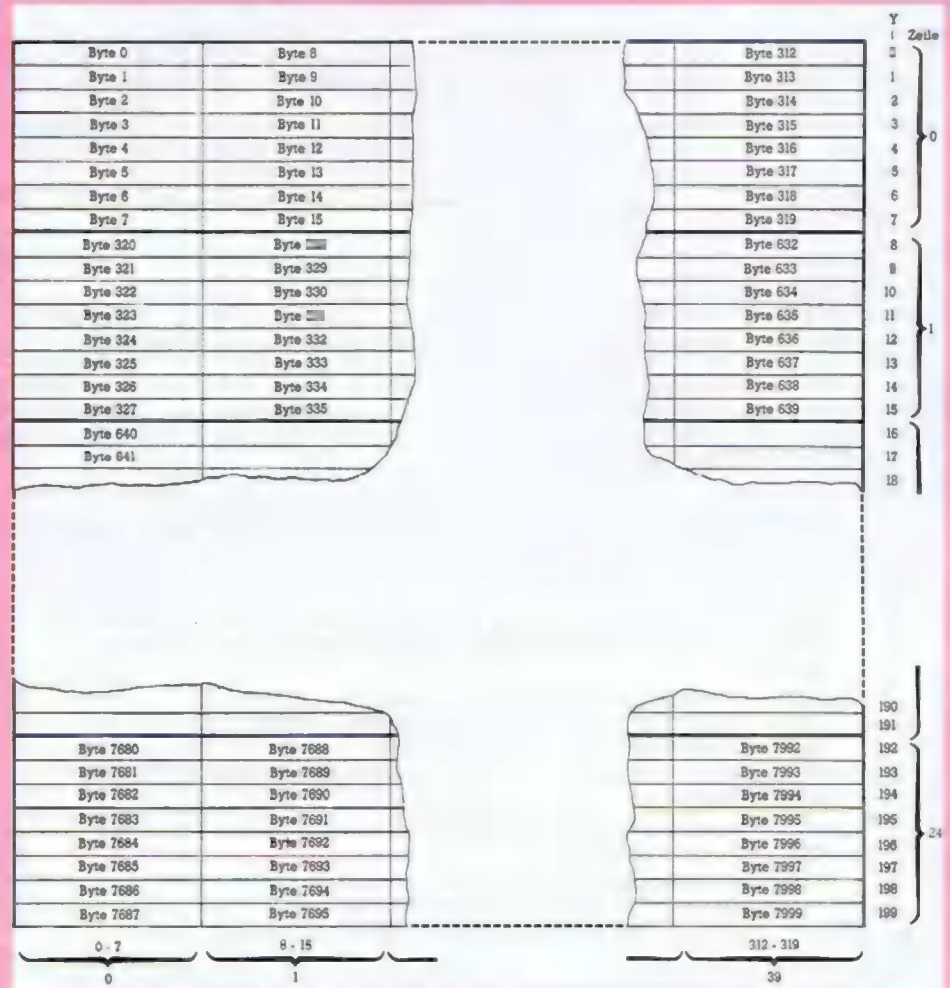


Bild 5. Zusammenhang zwischen Bildschirm und Bit-Map

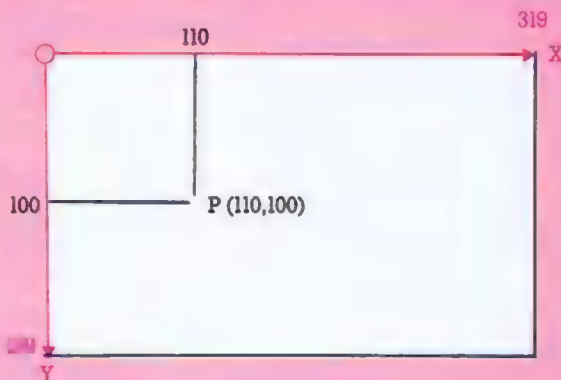


Bild 6. Bildschirmkoordinaten bei hochauflösender Grafik

Spalte 13	
	Byte 3944
	Byte 3945
	Byte 3946
	Byte 3947
	Byte 3948
	Byte 3949
	Byte 3950
	Byte 3951

Bild 7. Hier wird die Byte-Adresse zum Punkt (110,100) ermittelt ...

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

Byte 3948

Bild 8. ... und hier die Bit-Adresse

Falls Sie vorhin beim Anschalten des Mehrfarb-Modus nicht die normale Zeichenfarbe (Code 14), sondern eine zwischen 0 und 7 vorliegen hatten, dann haben Sie keine Veränderung bemerkt.

Und damit sind wir bei der zweiten Voraussetzung: Es kommt nämlich auch noch auf den Inhalt jeder Bildschirmfarbspeicherstelle an (55296 bis 56295). Erst wenn für eine solche Zelle das Bit 3 gesetzt ist (= 1), findet man Zeichen im dazugehörigen Bildschirmplatz in der Mehrfarbdarstellung. Sehen wir uns dazu noch einige Farbcodes an:

1 = bin. 0001

7 = bin. 0111

Bit 3 ist gelöscht

8 = bin. 1000

15 = bin. 1111

Bit 3 ist gesetzt

Der Farbcode bewirkt also zweierlei:

- Er gibt an, ob ein Zeichen im Mehrfarbmodus dargestellt wird, und
- er verleiht den Bit-Paaren »11« die Farbe.

Zur Demonstration können Sie ja mal folgendes Programm eingeben:

```

10 PRINT CHR$(147)
20 POKE 53281,6:POKE 53282,1:POKE 53283,0:REM LADEN DER 3 FARBREGISTER
30 POKE 53270,PEEK(53270)OR 16:REM ANSCHALTEN DES MEHRFARBEN-
  MODUS
40 POKE 646,0:REM ZEICHENFARBE AUF 0(BLK)
50 PRINT CHR$(17)"JETZT IST DER MEHRFARBENMODUS".PRINT
  "INGESCHALTET"
60 PRINT CHR$(17)"IN DIE BILDSCHIRMFARBZEILEN IST ABER DIE";PRINT"0
  EINGEGEBEN"
70 PRINT CHR$(17)"DESWEGEN IST DIE ZEICHENDARSTEL-
  LUNG";PRINT"NORMAL"
80 POKE 646,14:REM ZEICHENFARBE AUF 14(HBLU)
90 PRINT CHR$(17)"DER FARBCODE IST JETZT 14"
100 PRINT CHR$(17)"DER MEHRFARBEN-MODUS IST SICHTBAR"
110 END
120 POKE 53270,PEEK(53270)AND 239:REM AUSSCHALTEN DES MEHRFARBEN-
  MODUS
130 END

```

Wenn dieses Programm mit Zeile 110 beendet ist, befinden Sie sich weiterhin im Mehrfarbenmodus und können nach Herzenslust durch POKE 646, »Zeichenfarbe«, oder mit »CTRL« beziehungsweise »COM-MODORE«-Taste und einer Zifferntaste die Zeichenfarbe ändern und den Effekt beobachten. Übrigens ist der Cursor unterhalb der READY-Meldung noch da — was Sie durch eine Änderung der Zeichenfarbe sofort feststellen können. Sollten Sie genug von diesem Modus haben, dann geben Sie ein CONT »RETURN« und das Programm schaltet wieder den Normalmodus ein.

Sollten Sie zufällig mal durch PRINT PEEK(53281),PEEK(53282), PEEK(53283)

in die Hintergrundfarbregister sehen, dann werden Sie dort nicht — wie im Programm veranlaßt — die Ziffern 6,1,0 sehen, sondern 246,241,240. Das liegt daran, daß die Bits 4 bis 7 dieser Register für die Farbgebung nicht benutzt werden und vom Betriebssystem auf 1 gehalten werden. Die Binärzahl 1111 0000 entspricht dem Dezimalwert 240. Deswegen muß man zur Farbcodezahl diese 240 addieren, um den Speicherinhalt zu erhalten.

Wieder eine andere Art der Interpretation findet man beim Modus für erweiterte (oder veränderte) Hintergrundfarben. Damit kann man zum

(nicht das Zeichen selber) auf den Farbcode, der in 53282 steht. Ähnlich verläuft es mit dem reversen A (Bildschirmcode 129) und dem reversen geschifteten A (Bildschirmcode 193). In die Register 53281 bis 53284 können alle Farbcodes eingegeben werden.

Damit der Computer die Codes in dieser besonderen Weise liest, muß ihm das angezeigt werden, indem man im Byte 53265 das Bit 6 auf 1 setzt. Das sollten Sie schon können: POKE 53265,PEEK(53265)OR 64

Das Abschalten dieses Modus geschieht durch Löschen des gleichen Bits:

Bit-Paar zeigt auf:			
Bit-Paar	Register #	Adresse	Bedeutung
00	33	53281	Hintergrundfarbe Nr. 0
01	34	53282	Hintergrundfarbe Nr. 1
10	35	53283	Hintergrundfarbe Nr. 2
11	bringt die Farbe, die in Bit 0 bis 2 des Bildschirmfarbspeichers vorhanden ist		

Tabelle 1. Bedeutung von Bit-Kombinationen im Mehrfarbenmodus

POKE 53265,PEEK(53265)AND 191

Die Zeichenfarbe bleibt von der ganzen Umstellung unberührt. Man kann sie auf die nun schon bekannte Weise jederzeit ändern.

Warum kann man eigentlich in diesem Modus nur 64 Zeichen darstellen? Die Antwort darauf liegt darin, daß nur die Bits 0 und 5 für den Zeichencode zur Verfügung stehen, also

von 00 0000 ≙ »@«

bis 11 1111 ≙ »?«

mit Hintergrundfarbe aus 53281

Jede weitere Erhöhung beeinflusst die Bits 7 und 6, wird aber in den unteren 6 Bits nicht mehr wahrgenommen. Dort beginnt einfach die Zählung wieder ab 0:

(1)00 0000 = 64,

das bedeutet Zeichen 0 (@) mit Hintergrundfarbe aus 53282.

Ganz nett wäre es ja eigentlich, wenn man sowohl den Modus mit erweiterten Hintergrundfarben als auch dem Mehrfarbzeichenmodus miteinander kombinieren könnte. Jeder Versuch, das zu tun, endet mit einem schwarzen Bildschirm ohne jegliches Zeichen. An diese Möglichkeit haben die Schöpfer des VIC-II-Chip offenbar nicht gedacht.

Betrachten Sie jetzt doch einmal das nachfolgende Demonstrationsprogramm. Vor der Eingabe schalten Sie bitte durch gleichzeitiges Drücken der »COMMODORE«- und

Beispiel ein grünes Zeichen mit rotem Hintergrund auf einem gelben Bildschirm darstellen. Hier wird der Bildschirmcode des Zeichens auf eine vom Normalmodus unterschiedliche Weise gelesen. Bit 7 und Bit 6 der Bildschirmcodezahl geben nämlich jetzt einen Zeiger auf Farbregister ab. Das geschieht in der Weise, wie in der Tabelle 2 gezeigt wird.

Es zeigt sich, daß nur 64 Zeichen dargestellt werden können — also keine Grafikzeichen. Verwendet man das normale A (Bildschirmcode 1), dann zeigen die Bits 7 und 6 auf das normale Hintergrundregister 53281. Verwendet man das geschiftete A (Bildschirmcode 65), verändert sich die Hintergrundfarbe des A


```

5 poke chr$(147)
10 poke 53281,6:poke 53282,0:poke 53283,7:poke 53284,10:rem laden der farbregi-
ster
20 poke 53265,peek(53265) or 64:rem anschalten des modus
30 print chr$(17)"der modus ist eingeschaltet aber der":print"zeichencode ist
kleiner als 64"
40 print chr$(17)"JETZT WIRD MIT GESHIFTETEN ZEICHEN":print"GEDRUCKT"
50 print chr$(17)chr$(18)"der druck mit reversed zeichen hat die"
60 print chr$(18)"hintergrundfarbe von register 53283"
70 print chr$(144)chr$(17)chr$(18)"REVERSED UND GESHIFTETE ZEICHEN"
80 print chr$(154):end
90 poke 53265,peek(53265)and 191
100 end

```

»SHIFT«-Tasten den Kleinschriftmodus ein.

Ebenso wie beim vorigen Programm können Sie nach dem ersten END in Zeile 80 noch mit den verschiedenen Kombinationen von Zeichenfarbe und Zeichencode herumexperimentieren. Wenn Sie genug davon haben, erreicht ein CONT »RETURN« die Rückkehr in den Normalzustand.

Auch dieser Modus ist mit selbstdefinierten Zeichen ebenso verwendbar wie mit den ROM-Zeichen.

Wir betreten Dornröschens Schloß: Das Bit-Mapping-Prinzip

Es ist soweit! Wir stehen an der Schwelle zur hochauflösenden Grafik. Wenn wir bisher mit Zeichen aller Art auf dem Bildschirm gearbeitet haben, waren diese immer durch acht Bytes definiert: entweder durch das Zeichen-ROM oder durch einen von uns erstellten alternativen Zeichensatz im RAM. Jeder Aufruf eines Zeichencodes füllte einen Bildschirmspeicherplatz, zum Beispiel 1024 mit diesem Zeichen, also mit 8 Byte. Zugriff auf einzelne Byte hatten wir nur beim Selbstdefinieren, aber dieser Zugriff legte dann eben in der Anwendung auch wieder 8 Byte auf einmal fest. Wenn es nun eine Möglichkeit gäbe, ständig Zugriff auf einzelne Bytes zu haben und die Punktmuster jedes Bytes verändern zu können, dann hätten wir ganz andere Aussichten (Bild 3).

Überlegen wir mal: Statt 1000 Byte (40 x 25) hätten wir dann 8000 Byte (40 x 25 x 8) zu je 8 Bit, also insgesamt 64000 Bildpunkte, die ansprechbar wären! Genau das geschieht beim sogenannten Bit-Mapping. Der Computer legt sich gewissermaßen eine Landkarte dieser 8000 Byte an und bildet ihren Inhalt auf dem Bildschirm ab. Auf dem Bildschirm kön-

		Beispiel »A«			Hintergrundfarbregister	
Zeichen	Codebe- reich	Code	Code, binär	Bit Bit		
		dezi- mal	76543210	6 7	Nr.	Adresse
Normale Zei- chen	0- 63	1	00000001	0 0	33+0	53281
(SHIFT)-Zei- chen	64-127	65	01000001	0 1	33+1	53282
Reversed Zei- chen	128-191	129	10000001	1 0	33+2	53283
Reversed (SHIFT)-Zeichen	192-255	193	11000001	1 1	33+3	53284

Tabelle 2. Bildschirmcode und Mehrfarbenmodus

Bit 3 von 53272	Ort der Bit-Map (Adressenbereich)
0	0 - 7999
1	8192 - 16191

Tabelle 3. Mögliche Bereiche des Bit-Map

nen wir mit dieser Karte nun (siehe Bild 3) in der Senkrechten (y) 25 x 8 = 200 Positionen und in der Waagerechten (x) 40 x 8 = 320 Positionen ansprechen. Immer dann, wenn ein Bit in der Bit-Map gesetzt (= 1) ist, erzeugt das automatisch das Aufleuchten eines der 64000 Bildpunkte auf dem Bildschirm. Ist ein Bit gelöscht (= 0), dann ist der Bildpunkt leer (siehe Bild 4).

Dornröschen erwacht

Das Anschalten dieses Bit-Map-Modus ist der Kuß, den wir Dornröschen geben. Seien Sie nicht vorzeitig enttäuscht. Dornröschen hat 1000 Jahre geschlafen und wir müssen mit viel Geduld ihre Fähigkeiten nach und nach entwickeln helfen. Wenn Dornröschen jetzt erwacht, ist es hilflos wie ein neugeborenes Kind. Nach dieser Warnung also gehen wir so vor: Im VIC-II-Chip Register 53265 setzen wir Bit 5 auf 1, indem wir den Befehl

POKE 53265,PEEK(53265)OR 32 eingeben. Danach sieht der Bildschirm allerdings recht merkwürdig aus. Das ist auch kein Wunder, denn wie gesagt wird jetzt der Inhalt der Bit-Map dargestellt, und wir haben dem Computer noch nicht mitgeteilt, wo die Bit-Map zu finden ist. Das geschieht durch Setzen des Bit 3 des VIC-II-Registers mit der Adresse 53272 auf den Wert 0 oder 1. Dabei gelten die Zusammenhänge nach Tabelle 3.

Der normale Inhalt von Byte 53272 ist 21 (das kennen wir noch aus der letzten Folge). Binär ausgedrückt ist das: 0001 0101

Bit 3 ist also 0. Alles, was in diesem Bit-Map-Speicherbereich von 0 (Zeropage!) bis 7999 (mitten im Basic-RAM) existiert, wurde beim Anschalten des Bit-Map-Modus abgebildet. Und weil die Zeropage laufend vom Betriebssystem für seine ständigen Arbeiten verwendet wird, sind im oberen Bereich unseres seltsamen Bildes auch flimmernde Stellen zu sehen — wo also laufend Bits gesetzt und gelöscht werden. Sehen Sie sich Ihren Bildschirm mit diesem merkwürdigen Abbild noch etwas genauer an. Etwa von der Mitte an sehen Sie Zeichen abgebildet. Dort liegt die Speicherzelle 4096, von der an das Geisterbild der Zeichen (siehe Folge 1) liegt. Wir sehen also Gespenster!

Nun sollten Sie mit »RUN/STOP« und »RESTORE« wieder den Normalzustand herstellen. Dann schalten Sie mit folgenden Programmen den Hochauflösungsmodus und die spezielle Bit-Map ab 8192 an:

```
10 PRINT CHR$(147):POKE53265,
```



```
PEEK(53265)OR 32
20 POKE 53272,PEEK(53272)OR 8
Der letzte Befehl besorgt das Setzen des Bit 3 auf 1. Außerdem fügen wir vorsorglich noch die Rückkehr in den Normalmodus an:
110 POKE 53265,PEEK(53265)AND 223
Löschen des Bit 5 in 53265,
120 POKE 53272,PEEK(53272)AND 247
Löschen des Bit 3 in 53272.
```

Dazu bauen wir noch eine GET-Abfrage ein:

```
100 GET A$:IF A$ = " " THEN 100
```

Starten Sie jetzt mit RUN. Die Bitmap ab 8192 wird auf dem Bildschirm gezeigt. Das ist leerer Speicher, so wie wir ihn in Form von Zahlen (Blöcken von 0 und 255) schon in der 1. Folge kennengelernt haben.

Die ersten Schritte

Sie erinnern sich vielleicht auch daran, daß manchmal, wenn auch selten, andere Zahlen eingestreut waren. Deswegen sieht man hier auch nicht immer ganz schwarze oder ganz helle Blöcke, sondern häufig welche mit Einsprengseln. Überall dort, wo ein Byte der Bitmap den Wert 255 hat, steht ja binär 1111 1111, das heißt 8 Bildpunkte auf dem Bildschirm sind angeschaltet. Wir sehen schon, wir müssen den Speicher noch löschen, also überall 0 hinschreiben:

```
30 B=8192:FOR I=0 TO 7999:POKE I+B,0:NEXT I
```

Wenn Sie diese Programmzeile noch einfügen und dann mit RUN starten, können Sie dem Computer bei der Arbeit zusehen.

Jetzt kommt Farbe ins Bild

Vermutlich sind Sie jetzt enttäuscht. Ein leerer schwarzer Bildschirm ist nicht gerade eindrucksvoll. Deswegen wollen wir nun noch zwei Dinge lernen: Wie kommt Farbe ins Bild und wie bekomme ich Punkte auf den Bildschirm?

Wenn Sie vor der Zeile 100 im Programm einen Stop-Befehl einfügen, tauchen nach der Abarbeitung der ersten Zeilen plötzlich farbige Quadrate auf dem Bildschirm auf. Die Antwort auf die Frage nach der Herkunft dieser Quadrate birgt für uns zugleich die Möglichkeit der Farbgebung. Geben Sie nun ein: CONT

»RETURN«. Auch das erscheint als Folge farbiger Quadrate. Jetzt wartet der Computer auf einen Tastendruck, der auf dem Bildschirm folgendes erscheinen läßt:

```
BREAK IN 40
```

```
READY
```

```
CONT
```

```
READY
```

```
□
```

Der Mehrfarbenmodus wird entschleiert

Genau an derselben Stelle waren im HIRES-Modus die farbigen Quadrate zu sehen. Die Farbe wird also anscheinend vom Inhalt der Bildschirmspeicherzellen gesteuert. Das B von BREAK hat einen Bildschirmcode von 2. Zwei ist auch der Farbcode für Rot. Falls Sie einen Farbmonitor Ihr eigen nennen, sehen Sie im Hochauflösungsmodus an der Stelle des B ein rotes Quadrat. Des Rätsels Lösung sieht so aus: Die Farbe des Hochauflösungsbildes wird von 8x8-Bit-Feldern gesteuert, die den Bildschirmspeicherzellen entsprechen. Die Steuerung ist also vom Inhalt des Bildschirmspeichers abhängig, nicht vom Inhalt des Farbspeichers. Schreiben wir zum Beispiel in Zeile 1024 eine 1, dann erhalten wir im Hochauflösungsmodus ein weißes Feld in der linken oberen Bildschirmecke. Nun wollen wir in diesen Bereich auch noch eine kleine Linie zeichnen: Löschen Sie bitte zunächst mal den STOP-Befehl und fügen Sie ein:

```
40 POKE 1024,1:POKE 8193,255
```

Nach dem Starten erhalten wir — wie erwartet — das weiße Feld links oben und darin einen schwarzen Strich. Um offen zu sein — ganz so erwartet war's ja nicht. Denn woher hätten wir wissen können, daß der Strich schwarz ist? Sehen wir uns den Inhalt der Speicherzelle 1024 genauer an. Im binären System geschrieben steht dort jetzt:

```
0000      0001
```

```
MSN      LSN
= 0      = 1
```

N = Nibble (4 Bit)

Das legt die Vermutung nahe, daß die Hintergrundfarbe durch das LSN (hier 1 = weiß) und die Punktfarbe durch das MSN (hier 0 =

schwarz) gesteuert wird. Probieren wir es einfach mal aus. Wir tauschen das ganze mal um und schreiben in 1024 den folgenden Wert:

```
0001      0000
```

```
MSN      LSN
= 1      = 0
(dezimal 16)
```

Also ändern wir in der Zeile 40 den ersten POKE-Befehl um in POKE 1024,16. Es klappt! Nach dem Starten (wie immer in diesem Programm nach einigem geduldigen Warten) erscheint ein weißer Strich auf schwarzem Grund. Nun können wir dieses triste Schwarz gezielt entfernen und von vornherein bestimmen, welche Farbe unsere Punkte bekommen sollen und welche der Hintergrund, indem wir in die Bildschirmzellen den Dezimalwert unserer Binärzahl eingeben. Damit Sie nicht jedesmal umrechnen müssen, hier eine nützliche Formel:

Farbkennziffer (F) = Zeichenfarbe (ZF) x 16 + Hintergrundfarbe (HF)

Zum Ausprobieren ändern wir unser Programm um:

```
5 INPUT "ZF,HF = ";ZF,HF
```

```
In Zeile 40 schreiben wir jetzt neu:
40 F=ZF*16+HF:FOR I=1024 TO 2023:POKE I,F:NEXT I
```

```
50 POKE 8193,255
```

Wenn Sie beispielsweise nach dem Starten für die Zeichenfarbe 14 und für die Hintergrundfarbe 6 wählen, können Sie mit etwas Geduld wieder dem Computer bei der Arbeit zusehen und dann einen hellblauen Strich auf blauem Grund finden.

Dornröschen lernt zeichnen: Punkte setzen im Hochauflösungs-Modus

Wir sind es jetzt leid, immer einen fast leeren Bildschirm anzusehen. Ob er nun schwarz oder blau ist: Wir wollen nun auch mal etwas darauf sehen! Die Schwierigkeit liegt gar nicht darin, daß wir nicht wüßten, wie wir Punkte auf den Bildschirm kriegen. Wir müssen ja nur ein Byte im Bereich von 8192 bis 16191 mit ein paar gesetzten Bits versehen, zum Beispiel durch POKE 12000,1, und schon sehen wir einen Punkt. Das Problem liegt vielmehr darin, daß wir einen Weg finden, gezielt Punkte an bestimmte Orte des Bild-

schirms zu setzen. Dazu müssen wir den Aufbau der Bit-Map kennen und und ihren Zusammenhang mit dem Bildschirm. Das Prinzip ist in Bild 5 gezeigt.

Die Speicherzellen der Bit-Map sind also angeordnet wie ein vollgeschriebener Bildschirm. Dort hatte ja jeder Buchstabe seine 8 Byte im Punktemuster. Aus der Anordnung ergibt sich außerdem, daß wir in der Y-Richtung 200 mögliche Positionen (0 bis 199) und in der X-Richtung 320 (0 bis 319) finden. Wir haben somit Bildschirmkoordinaten (Bild 6) und können Punkte definieren, zum Beispiel im Bild 6 den Punkt P mit den Koordinaten $X=110$ und $Y=100$.

Wir wissen nur noch nicht, in welchem Byte wir welches Bit setzen müssen, um diesen Punkt zu sehen. Betrachten wir dazu Bild 5 genauer. Die 25 Bildschirmzeilen und die 40 Spalten sind noch erhalten. Jeder solche Bildschirmplatz besteht aus acht Byte mit je acht Bit. Stellen wir zunächst einmal fest, um welche Zeile es sich handelt:

$Z = \text{INT}(Y/8)$

Beispiel: $Z = \text{INT}(100/8) = 12$
(Zeile 12)

Dann ermitteln wir die Spalte, in der unser Punkt liegt:

$S = \text{INT}(X/8)$

Beispiel: $S = \text{INT}(110/8) = 13$
(Spalte 13)

In unserem Beispiel muß der Punkt in Zeile 12 an Spaltenplatz 13 stehen. Weil in jeder Zeile 320 Byte (8x40) und an jedem Platz 8 Byte vorhanden sind, fängt unser Bildschirmfeld bei Byte Nr.

$Z*320 + S*8 = 12*320 + 13*8 = 3944$ an
(siehe Bild 7).

Nun müssen wir feststellen, um welches Byte in diesem Feld es sich handelt. Wir haben vorhin beim Feststellen der Zeile einfach gleich $\text{INT}(Y/8)$ berechnet. Wenn wir nun nur $Y/8$ berechnen, kommen wir in unserem Beispiel auf $Y/8 = 100/8 = 12.5$. Ziehen wir davon $\text{INT}(Y/8)$ ab: $Y/8 - \text{INT}(Y/8) = 12.5 - 12 = 0.5$ und multiplizieren das Ergebnis mit 8:

$80*(Y/8 - \text{INT}(Y/8)) = 8*0.5 = 4$

Genau das ist unser gesuchtes Byte: Nr. 4 von oben. Das sieht komplizierter aus, als es ist. Aus alledem ergibt sich jetzt, in welchem Byte der Bit-Map ein Bit zu setzen ist:

$\text{BYTE} = Z*320 + S*8 + 8*(Y/8 - \text{INT}(Y/8))$
In unserem Beispiel ist es dann also das Byte 3948 der Bit-Map. Nun müssen wir noch herauskriegen, welches Bit in diesem Byte zu setzen ist (siehe Bild 8).

Auch hier haben wir vorhin bei der Berechnung der Spalte einfach gerechnet $\text{INT}(X/8)$. Jetzt rechnen wir:

$X - 8*\text{INT}(X/8) = 110 - 8*\text{INT}(110/8)$

Das ergibt 6. Wenn wir das von 7 abziehen (Nummer des höchstwertigen Bit), erhalten wir:

$\text{BIT} = 7 - (X - 8*\text{INT}(X/8))$

Die Zauberformeln

In unserem Beispiel ist $\text{BIT} = 1$. Es muß also Bit 1 gesetzt werden. Damit haben wir jetzt folgende Formeln zur Verfügung:

$\text{BYTE} = \text{INT}(Y/8)*320 + \text{INT}(X/8)*8 + 8*(Y/8 - \text{INT}(Y/8))$

und

$\text{BIT} = 7 - (X - 8*\text{INT}(X/8))$

Den ersten Ausdruck für Byte kann man noch etwas vereinfachen:

```

10 POKE 52,255:POKE 52,31:POKE 55,255:POKE 56,31
20 ZF=0:HF=6:F=16*ZF+HF
30 DEFN A(X)=50*SIN(X/30)+100
40 PRINT CHR$(147)
50 POKE 53265,PEEK(53265)OR 32
60 POKE 53272,PEEK(53272)OR 8
70 B=8192:FOR I=0 TO 7999:POKE B+I,0:NEXT I
80 FOR I=1024 TO 2023:POKE I,F:NEXT I
90 FOR X=0 TO 319:Y=FNA(X)
100 BY=(X AND 504)+40*(Y AND 248)+(Y AND 7):BI=7-(X AND 7)
110 POKE B+BY,PEEK(B+BY) OR (2*BI):NEXT X
120 GET A$:IF A$="" THEN 120
130 POKE 53272,PEEK(53272) AND (255-8)
140 POKE 53265,PEEK(53265) AND (255-32)
150 PRINT CHR$(147)

```

$\text{BYTE} = 8*(39*\text{INT}(Y/8) + \text{INT}(X/8) + Y/8)$

Die genaue Speicherzelle ergibt sich durch Addieren der Bit-Map-Startadresse (8192) zu BYTE. Um also einen Punkt zu setzen, gibt man das Kommando:

POKE

$8192 + \text{BYTE}, \text{PEEK}(8192 + \text{BYTE}) \text{ OR } (2*\text{BIT})$

Wir ergänzen jetzt unser Programm. Die Zeile 50 wird neu:

$50 \text{ BY} = 8*(39*\text{INT}(Y/8) + \text{INT}(X/8) + Y/8): \text{BI} = 7 - (X - 8*\text{INT}(X/8))$

Dann folgt:

60 POKE B+BY,PEEK(B+BY) OR (2*BI)

Außerdem müssen wir natürlich Punktkoordinaten eingeben:

6 INPUT "X,Y=";X,Y

Dabei ist darauf zu achten, daß X zwischen 0 und 319, Y zwischen 0 und 199 liegt. Starten Sie mit RUN, geben Sie die Koordinaten ein und nach einiger Weile wird Ihr Punkt auf dem Bildschirm zu sehen sein.

Auf eines sollten Sie noch achten, wenn Sie die Bit-Map bei 8192 begin-

nen lassen (in der nächsten Folge werden wir andere Möglichkeiten kennenlernen): Schützen Sie die Bit-Map vor dem Überschreiben durch Basic! Der verbliebene Speicherplatz für Basic-Text, Variable, Arrays, Strings ist so knapp, daß Sie ganz schnell die Bit-Map überschreiben. Deswegen sollten Sie (siehe Folge 2) eingeben:

POKE 51,255:POKE 52,31:POKE 55,255:POKE 56,31

Es ist geschafft: Die erste Grafik

Alles, was wir bis jetzt über die Hochauflösungsgrafik gelernt haben, soll hier nochmal als kleines Demonstrationsprogramm zusammengefaßt werden. Also NEW eingeben und dann:

Dieses Programm setzt 320 Punkte nach der Form einer Sinus-Funktion. Zur Erläuterung sei noch bemerkt, daß in Zeile 100 eine etwas elegantere Lösung der Berechnung von BYTE und BIT angeführt ist. Sie können aber genauso gut die bisher verwendete benutzen. In Zeile 150 wird der Bildschirm nochmal gelöscht, denn bei der Eingabe der Farben haben wir ja die Bildschirmzellen mit einem Zeichencode belegt, der im Normalmodus immer unter dem Cursor sichtbar wäre.

Dornröschen ist also erwacht und hat die ersten zaghaften Schritte getan. Alles geht noch etwas langsam vor sich. Besonders das Löschen der Bit-Map (Programmzeile 70) erfordert geduldiges Abwarten — jedenfalls solange wir in Basic arbeiten. Für heute haben wir genug getan. Die nächste Folge wird ebenfalls der hochauflösenden Grafik gewidmet sein. Wir hauchen ihr weiterhin Leben ein.

(Heimo Ponnath)

Der VC20 als

Ob für das Familienalbum, zur Bebilderung von Zeitschriften oder für andere Zwecke — Fotografieren macht

Spaß. Das Auge kann sich aber nur dann an den festgehaltenen Motiven erfreuen, wenn die Bilder wirklich gelungen sind. Um das zu erreichen, muß vor dem Drücken des Auslösers richtige »Kopfarbeit« geleistet werden.

Ganz wichtig ist, daß die einzelnen Größen, wie Belichtungszeit, Tiefenschärfe usw. aufeinander abgestimmt sind. Die benötigten Werte lassen sich aus Tabellen ablesen oder müssen jedes Mal nach festgelegten Formeln berechnet werden.

Wolf Robrahn, von Beruf Fotograf, hat sich eine wesentlich bequemere Möglichkeit geschaffen: Umständliches Nachschlagen oder Ausrechnen der Werte erspart ihm sein »Fotolehrling«, der VC 20.

Vor etwas mehr als zwei Jahren fing alles recht »harmlos« an, als er sich entschloß: »Ich brauche einen Mikrocomputer, um Grundlagen der überall gefragten EDV-Kenntnisse zu erlernen.« Er kaufte einen ZX 81. Was er zu dem Zeitpunkt noch nicht ahnte, war, daß Mikrocomputer nur selten »keimfrei« sind. Mittlerweile hat sich nämlich herausgestellt, daß er sich mit dem Computer den Bazillus ins Haus geholt hatte, der auch bei ihm das weitverbreitete und bekannte »Computerfieber« auslöste. Es machte sich recht schnell bemerkbar, seine Frau und seine zwei Kinder trafen ihn immer häufiger in trauter Zweisamkeit mit dem Computer an. Doch die recht enthusiastisch angebaute Beziehung zum ZX 81 entpuppte sich sehr schnell als Strohfeuer: Nicht etwa, daß Wolf Robrahn von seinem »Fieber« geheilt war — im Gegenteil — die Symptome hatten sich verstärkt. Sein Kenntnisstand und seine Fähigkeiten waren inzwischen soweit gewachsen, daß ihm sein Abend- und Wochenend-Freund nicht mehr genügte. Er wollte mehr. Zwei Alternativen standen zur Wahl: Die Standardausführung auszubauen oder ein größeres Modell »mit richtiger Ta-



Bild 2. Brennweite 50 mm, Blende 16 = Belichtungszeit 2 ■

statur« zu kaufen. Der computerbegeisterte Fotograf entschied sich für letzteres und besitzt seit zwei Jahren einen VC 20. Noch lieber wäre ihm der größere Bruder — der C 64 — gewesen, doch in Anbetracht der damaligen Preise gab es keine Vereinbarung zwischen diesem Modell und dem Familienbudget.

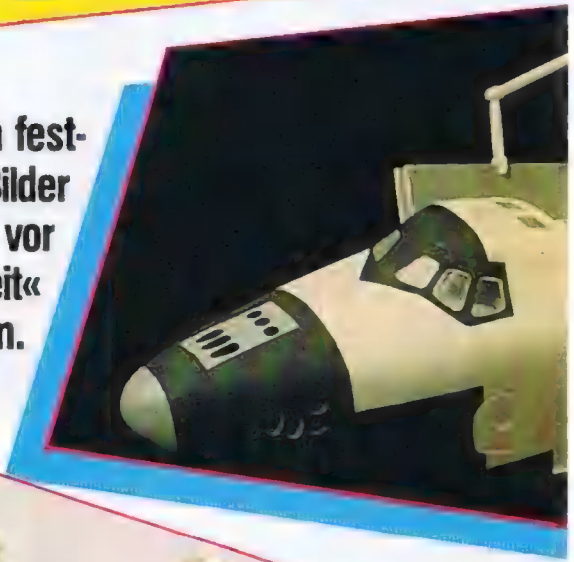
Am Anfang war das Spiel...

Anfangs probierte sich Wolf Robrahn im Programmieren von Computerspielen: Abenteuerspiele, Re-

aktionsspiele, Strategiespiele, Spiele mit oder ohne Grafik — zu jedem Thema entwickelte er Ideen und setzte sie mit dem VC 20 um. Doch die Spiellust erschöpfte sich bald. Es war an der Zeit, den Computer nutzbringend einzusetzen. Und was lag näher als ihn für Aufgaben heranzuziehen, die den Hobby- und Berufsfotografen entlasten.

Zwei »nutzbringende« Projekte sind seitdem schon realisiert: Die Einzelbild-Steuerung einer Super-8-Kamera mit dem VC 20 und das in diesem Artikel beschriebene

FOTOLEHRLING



HARLING



Bild 3. Brennweite 50 mm
mit Vorsatzlinse (Fish-
e) Blende 16 = Be-
lichtungszeit 1 s



Bild 1. Brennweite 50 mm, Blende 4 = Belichtungszeit 1/30 s

Programm zum fotografischen Fachrechnen. Momentan baut Wolf Robrahn nach den Schaltplänen von Commodore eine Modulbox für seinen »Fotolehrling«. Die benötigten Platinen stellt er »selbstverständlich« auf fotografischem Wege her. Dabei fertigt er zunächst eine Zeichnung an, die mit einem Strichfilm abfotografiert wird.

Auf einem Strichfilm gibt es nur schwarze und weiße Darstellungen, keine Grauwerte. Für die weitere Verarbeitung ist es unabdingbar, daß die feinen Linien »wirklich

schwarz« sind. Die Strichfilm-Aufnahme (Negativ) wird in der gewünschten Platinengröße auf eine Folie (Positiv) kopiert. Diese Folie ist die endgültige Vorlage: Sie wird auf die Platine gelegt, mit einer Glasplatte fest angedrückt und mit UV-Licht belichtet. Nach der Entwicklung lassen sich die auf die Platine übertragenen »Muster« ätzen. Auch das erledigt Wolf Robrahn selbst. Und noch ein Projekt hat der Hobby-Elektroniker geplant. Er möchte den VC 20 für eine totale Kamerasteuerung einsetzen: Der Computer soll

ihm bei Trickfilmaufnahmen assistieren, indem er die Filmkamera vor- und rückwärts, nach oben und unten sowie nach rechts und links bewegen kann. Der Meister möchte sich selbst von diesen Tätigkeiten entlasten, um sich ausschließlich auf die Motive konzentrieren zu können.

Und jetzt wird gerechnet

Die Bilder zeigen Aufnahmen, deren Werte mit dem Programm »Fotografisches Fachrechnen« ermittelt wurden. Bei Bild 1 war es Ziel, das Tonbandgerät im Hintergrund unscharf erscheinen zu lassen, während das Mikrofon im Vordergrund scharf abgebildet ist. Es handelt sich dabei in erster Linie um ein Tiefenschärfeproblem. Mit dem Programm wurde der Tiefenschärfereich ermittelt: Wolf Robrahn gibt nacheinander verschiedene Werte ein, bis einer der errechneten Schärfenbereiche zusagte (vergleiche auch das Anwendungsbeispiel weiter unten).

Fast bei jeder Mittelformat- und Kleinbildkamera gibt es eine am Blendenring eingravierte Tiefenschärfetabelle. Das Einstellen ist hier also kein Problem — man braucht nur richtig abzulesen (Bild 2). Schwieriger wird es, wenn man mit einer Vorsatzlinse arbeitet. Dann ist das Ermitteln der Tiefenschärfe am Blendenring nicht mehr möglich. Es muß gerechnet werden: Wieder ein Programm-Einsatz (Bild 3). Diese Beispiele sind nicht nur für Fotografien in Kleinbild- und Mittelformattechnik interessant, sondern ganz besonders auch bei der Großbildfotografie. Hier per Tastendruck die Bildgröße, Bildweite oder die korrekte Belichtungszeit zu ermitteln, ist eine wesentliche Erleichterung, die Wolf Robrahn vor allem bei Table-Top-Aufnahmen ständig nutzt.

Einen Haken hat das Ganze: Der VC 20 ist leider kein »Geländecomputer«, der in Feld, Wald und auf der Wiese genauso komfortabel einsetzbar wäre, wie auf dem heimischen Schreibtisch nahe der Steckdose. Bei Außenaufnahmen muß auch Wolf Robrahn die benötigten Werte ohne seine »Fotolehrling« ermitteln.

Hardwarevoraussetzungen für »Fotografisches Fachrechnen« ist ein VC 20 mit einer 8 KByte Speichererweiterung und einem Kassetteneinlesewerk. Die grafische Darstellung (Bild 4) veranschaulicht den allgemeinen Ablauf des Programms.

Der VC 20 als »Fotolehrling«

Nachdem es geladen und mit RUN gestartet wurde, erscheint auf dem Bildschirm das Menü (vergleiche Abbildung rechts). Über die Zahlentastatur wird der gewünschte Menüpunkt angewählt und die bekannten Größen werden im Dialog eingegeben. Hierbei gilt es einige Punkte zu beachten:

1. Bei Längenmaßen muß immer eine Größeneinheit angegeben werden. Hierzu ein Beispiel: Wird die Brennweite erfragt und man gibt nur »12« ein, erscheint die Fehlermeldung »EINHEIT VERGESSEN«. Richtig ist es, »12 CM« oder »120 MM« einzugeben. Wichtig ist, zwischen der Zahl und der Maßeinheit eine Leerstelle zu setzen. Wird eine Eingabe gefordert, deren Größe nicht bekannt ist, genügt auch hier nicht einfach eine »0« einzugeben — selbst in diesem Fall ist die Angabe einer Maßeinheit erforderlich. Um eine einheitliche Berechnungsgrundlage zu haben, werden automatisch alle Längenmaße in Millimeter umgerechnet.

2. Bei der Blende muß ein Wert der internationalen Blendenreihe eingegeben werden. Hiernach sind folgende Eingabe-Werte zulässig: 1.4; 2; 2.8; 4; 5.6; 8; 11; 16; 22; 32; 45; 64; 90 und 128.

Auch hierzu ein Beispiel: Wird die Blende abgefragt und man gibt »12.5« ein, erscheint die Fehlermeldung »KEINE GÜLTIGE BLENDE«. Erlaubt ist es, Blende »16« oder — da auch halbe Blenden einstellbar sind — »16*1/2« einzugeben. Das Sternchen zwischen 16 und 1/2 ist unbedingt zu setzen.

Erklärungen für die in dem Programm vorkommenden Fachbegriffe, wie Gegenstandsgröße, Bildweite und ähnliches können in dem Buch »Fotografie« von Kurt Dieter Solf, erschienen im Fischer Verlag, nachgeschlagen werden.

Ein kleines Beispiel

Ein Aufgabenbeispiel soll den Umgang mit dem Programm verdeutlichen:

Eine Briefmarke von 2 cm Breite und 3 cm Höhe soll formatfüllend mit einer Kleinbildkamera aufgenom-

```

1 rem fotografisches fachrechnen
2 rem
3 rem (c) 1984 by
4 rem wolf-d.robahn
5 rem beethovenstr. 16
6 rem 2200 elmshorn
7 rem
8 rem
10 Poke36879,174:Printchr$(144),chr$(14):Poke657,128
20 data1.4,2,2.8,4,5.6,8,11,16,22,32,45,64,90,128
99 rem **** menu ****
100 Print"☐☐☐ fotografisches          Fachrechnen          "
105 clr
110 Print"☐☐☐1. Berechnung der"
120 Print"☐☐☐Bildgröße"
130 Print"☐☐☐2. Berechnung der"
140 Print"☐☐☐Gegenstandsgröße."
150 Print"☐☐☐3. Verwendung von"
160 Print"☐☐☐Vorsatzlinsen"
170 Print"☐☐☐4. Berech. der Bildw."
180 Print"☐☐☐Brennw. & Entfern."
190 Print"☐☐☐5. Berechnung der"
200 Print"☐☐☐Belichtungszeit"
210 Print"☐☐☐6. Berechnung der"
220 Print"☐☐☐Tiefenschärfe"
230 Print"☐☐☐Code-Zahl eingeben"
239 rem **** code-eingabe ****
240 geta$
250 ifa$="1"then1000
260 ifa$="2"then2000
270 ifa$="3"then3000
280 ifa$="4"then4000
290 ifa$="5"then5000
295 ifa$="6"then6000
300 goto240
1000 Print"☐☐☐ *** BILDGRÖSSE *** "
1005 gosub11000
1010 Input"☐☐☐Bildweite☐"/a1$
1012 gosub12000:ifa1=1then1010
1015 gosub10000
1017 a=zw
1020 Input"☐☐☐Brennweite☐"/a1$
1022 gosub12000:ifa1=1then1020
1025 gosub10000
1027 f=zw
1030 Input"☐☐☐Gegenstandsgr.☐"/a1$
1031 gosub12000:ifa1=1then1030
1033 gosub10000
1034 g=zw
1036 Input"☐☐☐Entfernung☐"/a1$
1037 gosub12000:ifa1=1then1036
1040 gosub10000
1045 e=zw
1050 ifa<0andg<0andf<0thenb=(a*g)/e:goto1200
1060 iff<0andg<0andf<0thenb=(f*g)/(e-f):goto1200
1063 ifa=0andf<0then1080
1064 ifa=0andf=0then1080
1065 ifa<fthen60000
1066 ifa=fthen60500
1070 ifa<0andg<0andf<0thenb=((a-f)*g)/f:goto1200
1080 gosub61000
1085 clr
1090 goto1010
1200 b=int(100*b)/100
1210 ifb<100thenm$="mm"
1220 ifb=100thenb=b/10:m$="cm"
1230 iff=0thenf=int(100*((e*a)/(e-a))/100:Print"☐☐☐Brennweite"f"mm"
1240 ifa=0thena=int(100*((e*f)/(e-f))/100:Print"☐☐☐Bildweite"a"mm"
1250 ife=0thene=int(100*((a*f)/(a-f))/100:Print"☐☐☐Entfernung"e"mm"
1260 Print"☐☐☐BILDGRÖSSE"b,m$
1270 Print"☐☐☐RETURN☐☐ wenn weiter"
1280 geta$:ifa$<chr$(13)then1280
1290 goto100
2000 Print"☐☐☐ GEGENSTANDSGRÖSSE ***"
2005 gosub11000
2010 Input"☐☐☐Bildgröße☐"/a1$
2020 gosub12000:ifa1=1then2010
2030 gosub10000
2040 b=zw
2050 Input"☐☐☐Bildweite☐"/a1$
2060 gosub12000:ifa1=1then2050
2070 gosub10000
2080 a=zw
2090 Input"☐☐☐Entfernung☐"/a1$
2100 gosub12000:ifa1=1then2090
2110 gosub10000
2120 e=zw
2140 Input"☐☐☐Brennweite☐"/a1$
2150 gosub12000:ifa1=1then2140
2160 gosub10000
2170 f=zw
2180 ifa<0andb<0andg<0then9=(e*b)/a:goto2300
2185 ifa=0andf<0then2220
2190 ifa=0andf=0then2240
2200 ifa<fthen60000
2210 ifa=fthen60500
2220 ife<0andb<0andf<0then9=((e-f)*b)/f:goto2300
2230 iff<0andb<0andf<0then9=((f*b)/(a-f)):goto2300
2240 gosub61000
2250 clr:goto2010
2300 9=int(100*9)/100
2310 if9<100thenm$="mm"
2320 if9=100then9=9/10:m$="cm"
2330 iff=0thenf=int(100*((e*a)/(e+a))/100:Print"☐☐☐Brennweite"f"mm"

```

1. Berechnung der Bildgröße
2. Berechnung der Gegenstandsgröße
3. Verwendung von Vorsatzlinsen
4. Berechnung der Bildweite, der Brennweite und Entfernung
5. Berechnung der Belichtungszeit
6. Berechnung der Tiefenschärfe

Hauptmenü

Basic-Listing für fotografisches
Fachrechnen


```

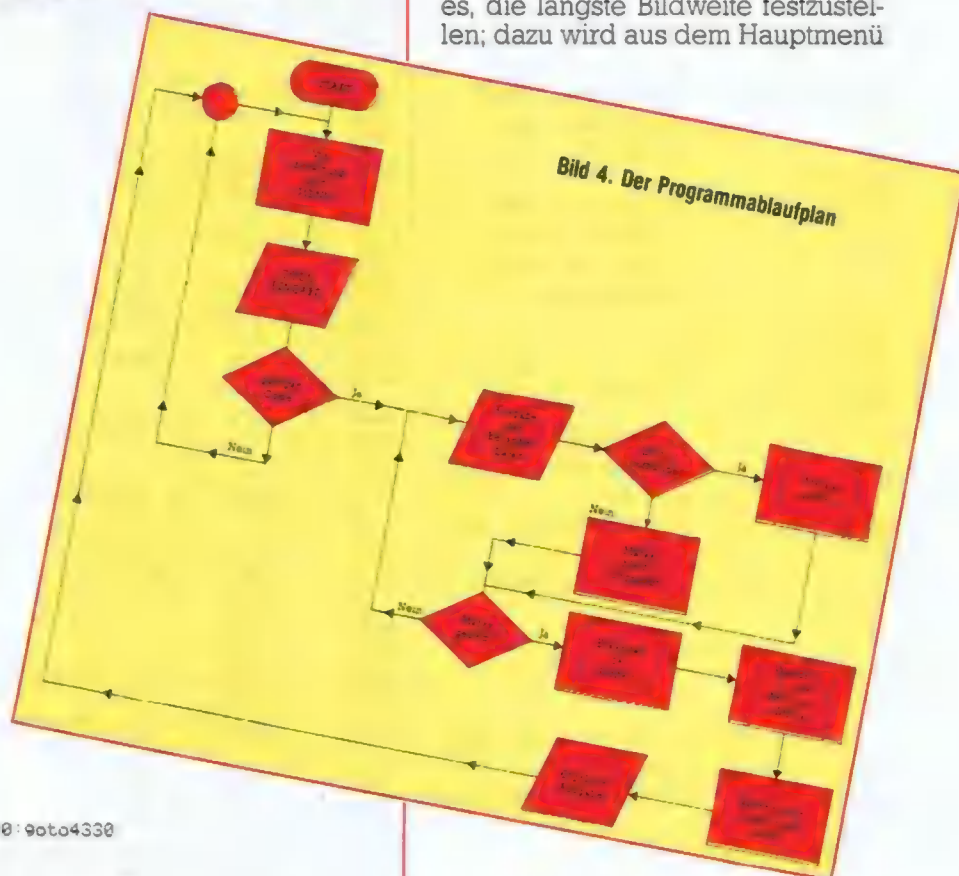
2340 ifa=0thena=int(100*((e*f)/(e-f))/100:Print"Bildweite"a"mm"
2350 ife=0thene=int(100*((a*f)/(a-f))/100:Print"Entfernung"e"cm"
2360 Print"GEGENSTANDSGR."g;ms
2370 Print"RETURN wenn weiter"
2380 geta$:ifa$<>chr$(13)then2380
2390 goto100
3000 Print"*** VORSATZLINSEN ***"
3010 gosub1000
3020 input"Brennweite alt";a1$
3030 gosub1200:ifa1=1then3020
3040 gosub1000
3050 f1=zu/10
3060 input"Brennweite neu";a1$
3070 gosub1200:ifa1=1then3060
3080 gosub1000
3090 f2=zu/10
3100 input"Dioptrien";di
3105 iff1<>0andf2<>0thendi=(100/f2)-(100/f1):di=int(10*di)/10
3110 iff1<>0andf2<>0thendi=int(10*(100/f1))/10:di=int(10*di)/10
3120 iff1<>0andf2<>0thenPrint"DIOPTRIEN:"di:goto3150
3125 iff2=0thendi=0thendi=int(10*(100/f1))/10:Print"DIOPTRIEN:"di:goto3150
3126 iff1=0andf2=0thenf=int(10*(100/di)):Print"BRENNWEITE:"f"mm":goto3150
3130 iff1<>0thendi<>0thenf2=(100*f1)/(100+(di*f1)):f2=int(10*f2)
3140 iff1<>0thendi<>0thenPrint"BRENNWEITE NEU:"f2"mm"
3150 Print"RETURN wenn weiter"
3160 geta$:ifa$<>chr$(13)then3160
3170 goto100
4000 Print" Berechnung von A,f,E "
4010 gosub1000
4020 input"Bildgroesse";a1$
4030 gosub1200:ifa1=1then4020
4040 gosub1000
4050 b=zu
4060 input"Bildweite";a1$
4070 gosub1200:ifa1=1then4060
4080 gosub1000
4090 a=zu
4100 input"Brennweite";a1$
4110 gosub1200:ifa1=1then4100
4120 gosub1000
4130 f=zu
4140 input"Gegenstandsgr.";a1$
4150 gosub1200:ifa1=1then4140
4160 gosub1000
4170 g=zu
4180 input"Entfernung";a1$
4190 gosub1200:ifa1=1then4180
4200 gosub1000
4210 e=zu
4220 ifa<fanda<>0andf<>0then5000
4230 ifa=fanda<>0andf<>0then5050
4300 iff=0anda<>0anda<>0then4500
4310 iff=0anda<>0andb<>0andg<>0then4350
4320 iff=0andf<>0andb<>0andg<>0then4600
4330 ifa=0andf<>0andb<>0andg<>0then4700
4340 ifa=0anda<>0andf<>0then4720
4350 ifa=0andb<>0andf<>0anda<>0then4750
4360 ife=0andf<>0andb<>0andg<>0then4850
4370 ife=0anda<>0andf<>0then4950
4380 ife=0andg<>0andb<>0anda<>0then4870
4390 ifx=0thengosub5100:clr:goto4020
4450 Print"RETURN wenn weiter"
4460 geta$:ifa$<>chr$(13)then4460
4470 goto100
4500 f=(e*a)/(e+a):goto4650
4550 f=a/((b/g)+1):goto4650
4600 f=e/((g/b)+1)
4650 f=int(f):Print"BRENNWEITE"f"mm":x=1:f=0:goto4330
4700 a=f*((b/g)+1):goto4800
4720 a=(e*f)/(e-f):goto4800
4750 a=(b*e)/g
4800 a=int(a)/10:Print"BILDWEITE"a"cm":x=1:a=0:goto4360
4850 e=f*((g/b)+1):goto4900
4860 e=(a*f)/(a-f):goto4900
4870 e=(g*a)/b
4900 e=int(e)/10:Print"ENTFERNUNG"e"cm":x=1:e=0:goto4450
5000 Print" ** BELICHTUNGSZEIT **"
5005 gosub11000
5007 dimb(15)
5010 input"Blende";a1$a2$a3=a1$
5011 restore:forz=1to14
5012 readp:b(z)=p
5013 ifval(a1$)=Pthenzz=1
5014 next
5015 ifzz=0thenPrint"keine gueltige Blende!":goto5010
5017 gosub52000
5020 input"Brennweite";a1$
5030 gosub1200:ifa1=1then5020
5040 gosub1000
5050 f=zu
5060 input"Bildweite";a1$
5070 gosub1200:ifa1=1then5060
5080 gosub1000
5090 a=zu
5095 ifa<fthen5000
5100 input"Bel. Zeit";a1$a3=a1$
5110 ifleft$(a1$,2)="1/"thena1$=right$(a1$,len(a1$)-2):t=1/val(a1$):goto5200
5120 t=val(a1$)
5200 fa=(a*a)/(f*f)
5210 t=t*fa:fa=sqr(fa):ift<1thent$=str$(int(1/t)):t$=right$(t$,len(t$)-1)
5220 ift=1thent$=str$(int(10*t)/10)
5230 n=n/fa

```

Basic-Listing für fotografisches Fachrechnen (Fortsetzung)

men werden. Welche Vorsatzlinse ist notwendig, wenn eine Brennweite von 5 cm zur Verfügung steht. Der maximal einstellbare Nahbereich beträgt 40 cm.

Zu berechnen sind: Bildweite, neue Brennweite, Dioptrien-Wert (Brennkraft der Linse). Zunächst gilt es, die längste Bildweite festzustellen; dazu wird aus dem Hauptmenü



das Untermenü 4 angewählt. Gefragt wird nach:

BILD-GROESSE

Eingabe: 0 MM
(zur Berechnung nicht wichtig)

BILDWEITE

Eingabe: 0 MM
(Größe nicht bekannt, soll berechnet werden)

BRENNWEITE
GEGENSTANDS-GROESSE

Eingabe: 50 MM

ENTFERNUNG
Ergebnis:

Eingabe: 0 MM
(zur Berechnung nicht wichtig)

Eingabe: 40 CM
BILDWEITE = 5.7 CM

Der VC 20 als »Fotolehrling«

Die größte Bildweite beträgt also 57 Millimeter. Zur Berechnung der Brennweite wird abermals das Untermenü 4 angewählt. Gefragt wird wieder wie vorher:

BILD- GROESSE	Eingabe: 36 MM (Kleinbildformat)
BILDWEITE	Eingabe: 57 MM (wie vorher errechnet)
BRENN- WEITE	Eingabe: 0 MM (Größe nicht bekannt, soll errechnet werden)
GEGEN- STANDS- GROESSE	Eingabe: 3 CM
ENTFER- NUNG	EINGABE: 0 MM (soll berechnet werden)
Ergebnis:	BRENNWEITE = 25 MM ENTFERNUNG = 4.7 CM

Mit der neuen Brennweite läßt sich nun — über das Untermenü 3 — die gesuchte Vorsatzlinse berechnen.

Gefragt wird nach:

BRENNWEI- TE ALT	Eingabe: 50 MM
BRENNWEI- TE NEU	Eingabe: 25 MM
DIOPTRIEN	Eingabe: 0 (soll berechnet werden)
Ergebnis:	DIOPTRIEN = 20

Hiernach muß eine positive Vorsatzlinse von 20 Dioptrien Brechkraft benutzt werden, um die gestellte Aufgabe zu erfüllen. Wer die Rechnung überprüfen möchte, kann die Gegenprobe machen. Bei BRENNWEITE NEU muß hierbei »0 MM« und bei DIOPTRIEN »20« eingegeben werden.

(Wolf Robrahn/kg)

```

5240 forz=1to13
5250 ifn=bl(z)thenbl$=str$(bl(z)):goto5500
5260 ifn=bl(z)andn(bl(z+1))then90sub62500:goto5500
5270 next
5500 Print"*****Belichtung*****":Print"Blende: "a2$
5510 Print"Bel. Zeit: "t$ sec"
5520 Print"Blender..."
5530 Print"Blende: "bl$
5540 Print"Bel. Zeit: "a3$ sec"
5550 Print"*****RETURN wenn weiter"
5560 Getas: ifa$(chr$(13))then5560
5570 goto100
6000 Print"*****TIEFENSCHÄRFE*****"
6010 Print"Bitte geben Sie Ihr Filmformat ein!"
6020 Print"Bei Verwendung von Schmalfilmen:"
6030 Print"116 mm = 16"
6040 Print"135 Super 8 = s8"
6050 Print"Normal 8 = n8"
6055 Print"*****Beispiel:"
6056 Print"Format: 24x36 mm"
6057 Print"Format: 24x36 mm"
6060 input"Format: 24x36 mm";a1$a2$a3$
6065 ifa1$="s8"then6060
6070 ifa1$="n8"then6060
6080 ifa1$="16"then6060
6090 90sub12000:if9=1then6060
6100 a1$=left$(a1$,len(a1$)-2)
6105 a$=""
6110 forz=1to10
6120 a$=a$+mid$(a1$,z,1)
6130 ifmid$(a1$,z+1,1)="x"then6150
6140 next
6150 b$=right$(a1$,len(a1$)-(z+1))
6160 a1$=a$+right$(a2$,2)
6170 90sub10000
6180 a1$=zu
6190 a1$=b$+right$(a2$,2)
6200 90sub10000
6210 b1$=zu
6220 c=sqr((a1$a1)+(b1$b1)):sz=c/1500
6300 sz=int(1000*sz)/1000
6305 sz$=str$(sz):sz$=right$(sz$,len(sz$)-1)
6310 forz=1to10
6320 ifmid$(sz$,z,1)=chr$(69)then6340
6330 next
6335 goto6500
6340 sz$="."&left$(sz$,1)
6500 Print"*****Toleranz: "sz$ mm"
6505 forz=1to500:next
6510 a2$="":z=0:a$="":b$="":a1=0:b1=0:c=0:sz$=""
6511 Print"*****Wenn mit errechneter Toleranz einverstanden bitte nur RETURN"
6520 input"*****Toleranz: "sz$
6525 dimbl(15):restore:forz=1to14:readp:bl(z)=p:next
6529 rem ***** 2. menu *****
6530 Print"*****mögliche Berechnungsarten *****"
6540 Print"*****1. Tiefensch. bei"
6545 Print"*****Einstell. Unendl."
6550 Print"*****2. Tiefensch. bis"
6555 Print"*****Unendlich"
6560 Print"*****3. Tiefensch. von"
6565 Print"*****Unendl.Nahpunkt bis"
6570 Print"*****Unendl.Fernpunkt"
6575 Print"*****Code-Zahl eingeben"
6579 rem ***** code-eingabe *****
6580 Getas$
6585 ifval(a$)=1then7000
6590 ifval(a$)=2then7500
6595 ifval(a$)=3then8000
6599 goto6580
7000 Print"*****Einstellung UNENDLICH"
7010 input"*****Brennweite: "a1$
7015 90sub12000:if9=1then7010
7020 90sub10000
7025 f=zu
7030 input"*****Blende: "a1$:zz=0
7031 ifa1$="0"thenm=0:goto7400
7032 restore:forz=1to14:readp
7034 ifval(a1$)=pthenzz=1
7035 next
7040 ifzz=0thenPrint"keine gueltige Blende!":goto7030
7050 90sub62000
7055 np=((f*f)/(sz*sz*10))
7100 ifnp=1000thennp=int(np)/100 m$="m" goto7300
7110 ifnp=100thennp=int(np) m$="cm" goto7300
7120 ifnp<100thennp=int(10*np)/10 m$="cm"
7300 ifx=0thenPrint"*****Schärfte von "np,m$:Print"bis Unendlich"
7400 goto63000
7500 Print"*****Nahpunkt bis Unendlich"
7505 input"*****Brennweite: "a1$
7510 90sub12000:if9=1then7505
7515 90sub10000
7520 f=zu
7525 input"*****Blende: "a1$:zz=0
7530 ifa1$="0"thenm=0:goto7550
7532 restore:forz=1to14:readp
7533 ifval(a1$)=pthenzz=1
7535 next
7540 ifzz=0thenPrint"keine gueltige Blende!":goto7525
7545 90sub62000
7550 input"*****Nahpunkt: "a1$
7555 90sub12000:if9=1then7550
7560 90sub10000
7565 np=zu
7570 ifn=0then7700

```



```

7580 NP=(F*F)/(2*SZ*N*10)
7590 IF NP=100 THEN NP=INT(NP)/100:M$="m":GOTO 7600
7595 IF NP<100 THEN NP=INT(10*NP)/10:M$="cm"
7600 PRINT "Schaerfe von"NP,M$
7610 PRINT "bis Unendlich."
7620 PRINT "SEE INSTELLUNG"2*NP,M$
7630 GOTO 63000
7700 N=(F*F)/(2*SZ*NP)
7705 FOR Z=1 TO 13
7710 IF N=BL(Z) THEN BL$=STR$(BL(Z)):GOTO 7730
7715 IF N=BL(Z) AND N=BL(Z+1) THEN GOSUB 62500:GOTO 7730
7720 NEXT Z
7730 PRINT "Moetige Blende"BL$
7740 GOTO 63000
8000 PRINT "NAHPUNKT bis FERNPUNKT"
8010 INPUT "NAHPUNKT";A1$
8014 GOSUB 12000:IF A1=1 THEN 8010
8016 GOSUB 10000
8018 NP=NZ
8020 INPUT "FERNPUNKT";A1$
8025 GOSUB 12000:IF A1=1 THEN 8020
8030 GOSUB 10000
8035 FP=NZ
8040 INPUT "BRENNWEITE";A1$
8045 GOSUB 12000:IF A1=1 THEN 8040
8050 GOSUB 10000
8055 F=NZ
8060 IF N<0 AND FP<0 THEN 9000
8100 INPUT "Blende";A1$:ZZ=0
8110 RESTORE FOR Z=1 TO 14:READ P
8120 IF VAL(A1$)=P THEN ZZ=1
8130 NEXT Z
8140 IF ZZ=0 THEN PRINT "keine gueltige Blende!" GOTO 8100
8150 GOSUB 62000
8160 INPUT "SEE instellung";A1$
8165 GOSUB 12000:IF A1=1 THEN 8160
8170 GOSUB 10000
8180 E=NZ:E=E/10
8190 NU=(F*F)/(SZ*N*10)
8195 NP=(NU*E)/(NU*E)
8196 IF NU<0 THEN NU$="Unendlich" GOTO 8230
8200 FP=(NU*E)/(NU*E)
8210 IF NP<100 THEN NP=INT(10*NP)/10:M1$="cm"
8220 IF FP<100 THEN FP=INT(10*FP)/10:M1$="m"
8230 IF NP<100 THEN NP=INT(10*NP)/10:M$="cm"
8240 IF FP<100 THEN FP=INT(10*FP)/10:M$="m"
8250 PRINT "NAHPUNKT "NP,M$
8260 PRINT "FERNPUNKT "FP,M1$
8270 GOTO 63000
8300 E=(2*NP*FP)/(NP+FP)
8310 N=((F*F)*(FP-NP))/(2*SZ*NP*FP)
8320 FOR Z=1 TO 13
8325 IF N=BL(Z) THEN BL$=STR$(BL(Z)):GOTO 8345
8330 IF N=BL(Z) AND N=BL(Z+1) THEN GOSUB 62500:GOTO 8345
8340 NEXT Z
8345 IF N=100 THEN E=INT(E/100):M$="m":GOTO 8360
8350 IF N=100 THEN E=INT(E/10):M$="cm":GOTO 8360
8355 IF N<100 THEN E=INT(10*E)/10:M$="mm"
8360 PRINT "SEE INSTELLUNG"E,M$
8370 PRINT "Moetige BLENDE"BL$
8380 GOTO 63000
9999 REM *** wandeln in millimeter ***
10000 IF RIGHT$(A1$,2)="m" THEN NZ=VAL(A1$)*1000
10010 IF RIGHT$(A1$,2)="cm" THEN NZ=VAL(A1$)*10
10020 IF RIGHT$(A1$,2)="mm" THEN NZ=VAL(A1$)
10030 RETURN
11000 PRINT "Geben Sie folgende Werte ein:"
11010 RETURN
11999 REM *** ueberpruefen ob einheit ***
12000 A=0:IF RIGHT$(A1$,1)<>CHR$(76) THEN A=1:PRINT "Einheit vergessen"
12010 RETURN
50000 PRINT "Die Rechnung ist unzu-laessig, da die Abbildung vor der"
50010 PRINT "1. Brennweite liegen wurde."
50020 PRINT "RETURN wenn weiter"
50030 GET A$:IF A<>CHR$(13) THEN 50030
50040 GOTO 100
50500 PRINT "Die Berechnung ergibt UNENDLICH. Wenn moeg- lich statt Bildweite die ";
50510 PRINT "Entfernung ein- geben!"
50520 PRINT "RETURN wenn weiter"
50530 GET A$:IF A<>CHR$(13) THEN 50530
50540 GOTO 100
51000 PRINT "ungenuegende Angaben BITTE UEBERPRUEFEN"
51010 RETURN
51999 REM *** berechnung d. blende in dezimal ***
52000 N=VAL(A1$):IF RIGHT$(A1$,3)="1/2" THEN N=N*1.2
52010 RETURN
52499 REM *** umwandlung d. dezimalen blende ***
52500 X1=N-BL(Z):X2=BL(Z+1)-N:IF X1=X2 THEN 62600
52510 IF X2>X1 THEN 62700
52520 BL=BL(Z)*1.2:X1=N-BL
52610 IF X1>X2 THEN BL$=STR$(BL(Z+1)):GOTO 62800
52620 IF X2>X1 THEN BL$=STR$(BL(Z))+".1/2":GOTO 62800
52700 BL=BL(Z)*1.2:X2=BL-N
52710 IF X1>X2 THEN BL$=STR$(BL(Z))+".1/2":GOTO 62800
52720 IF X2>X1 THEN BL$=STR$(BL(Z))
52800 RETURN
63000 PRINT "Neu oder weiter"
63010 GET A$:IF A<>"n" AND A<>"u" THEN 63010
63020 IF A="n" THEN 100
63030 CLR:GOTO 6520
READY.

```

Basic-Listing für fotografisches
Fachrechnen (Schluß)

Variablenliste

A	= BILDWEITE
A1	= SEITENLAENGE 1
B	= BILDGROESSE
BL	= SEITENLAENGE +
BL(X)	= BLENDENREIHE
C	= BILDDIAGONALE
DI	= DIOPTRIEN
E	= ENTFERNUNG
F	= BRENNWEITE
F1	= BRENNWEITE ALT
FA	= FAKTOR
FP	= FERNPUNKT
G	= GEGENSTANDS- GROESSE
N	= ERRECHNETE BLENDE
NP	= NAHPUNKT
NZ	= NAHPUNKT UNEND- LICH
P	= BLENDENVARIABLE
SZ	= SCHAEFERTOLERANZ
T	= ZEITVARIABLE
X	= KONTROLLVARIABLE
X1	= HILFSVARIABLE FUER BLENDENBERECHNUNG
X2	= DITO
Z	= SCHLEIFENVARIABLE
ZW	= ZWISCHENVARIABLE
ZZ	= KONTROLLVARIABLE
AS	= HILFSVARIABLE
A1\$	= EINGABEVARIABLE ZUR WEITEREN AUFBE- REITUNG
A2\$	= HILFSVARIABLE FUER BLENDE
A3\$	= HILFSVARIABLE FUER ZEIT
B\$	= HILFSVARIABLE
BL\$	= BLENDENWERT ALS STRING
M\$	= VARIABLE FUER MM # CM ODER M
M1\$	= DITO
SZ\$	= SCHAEFERTOLERANZ ALS STRING
T\$	= ZEIT ALS STRING

Wir suchen die Anwendung des Monats

Anwendung des Monats, was ist das? Nun, Sie haben einen Commodore 64 oder einen VC 20 und versuchen diesen irgendwie sinnvoll einzusetzen. Unter einer sinnvollen Anwendung versteht die 64'er Redaktion alles, was beispielsweise Programme im häuslichen Bereich bewirken. Es kann sich dabei um die Berechnung der Benzinkosten für Ihren Wagen handeln, um ein eigenes Textverarbeitungsprogramm gehen, sich um die Verwaltung Ihrer Tiefkühltruhe drehen oder ein ausgeklügeltes Telefon- und Adreßregister sein.

Setzen Sie Ihren VC 20/C 64 mehr oder weniger beruflich ein? Auch, oder vor allem, das ist eine sinnvolle Anwendung. Sie führen die Lohn- und Gehaltsabrechnung, Ihre Lagerverwaltung, die Bestellungen auf einem Commodore-Heimcomputer durch? So spezielle Anwendungen wie die Berechnung der Statik von selbstgezimmernten Regalen, von Klimadiagrammen oder Vokabellernprogrammen für den Schulunterricht oder die Zinsberechnung bei Krediten sind ebenfalls Themen, die mehr als konkurrenzfähig sind.

Uns ist die Anwendung des Monats

500 Mark

Schreiben Sie uns, was Sie mit Ihrem Computer machen:

Redaktion 64'er, Aktion: Anwendung des Monats, Hans-Pinsel-Str. 2, 8013 Haar bei München.

Inserentenverzeichnis

Ariola 5
Atari 12

Birkhäuser 95

Commodore 58/59
Computer Buchla-
den 38/39,
110-113, 148
Computer Plus Soft
45, 90

Data Becker
2, 16/17, 76/77

FRIWA 94

Interface AGE 88
IWT Verlag 89

Kingsoft 88

Lindy 95

Roos 93

S + S Software
J. Schlüter 91
Sybex 29, 75

Wiesemann 90
WS-Werbeteam 96

Impressum

Bezugspreise: Das Einzelheft kostet DM 6,-. Der Abonnementspreis beträgt im Inland DM 72,- pro Jahr für 12 Ausgaben. Darin enthalten sind die gesetzliche Mehrwertsteuer und die Zustellgebühren. Der Abonnementspreis erhöht sich um DM 18,- für die Zustellung im Ausland, für die Luftpostzustellung in Ländergruppe 1 (z.B. USA) um DM 38,-, in Ländergruppe 2 (z.B. Hongkong) um DM 58,-, in Ländergruppe 3 (z.B. Australien) um DM 68,-.

Druck: St. Otto-Verlag, Bamberg

Urheberrecht: Alle im »64'er« erschienenen Beiträge sind urheberrechtlich geschützt. Alle Rechte, auch Übersetzungen, vorbehalten. Reproduktionen gleich welcher Art, ob Fotokopie, Mikrofilm oder Erfassung in Datenverarbeitungsanlagen, nur mit schriftlicher Genehmigung des Verlages. Anfragen sind an Klaus Buck zu richten. Für Schaltungen und Programme, die als Beispiele veröffentlicht werden, können wir weder Gewähr noch irgendwelche Haftung übernehmen. Aus der Veröffentlichung kann nicht geschlossen werden, daß die beschriebenen Lösungen oder verwendeten Bezeichnungen frei von gewerblichen Schutzrechten sind. Anfragen für Sonderdrucke sind an Klaus Buck zu richten.

© 1984 Markt & Technik Verlag Aktiengesellschaft,
Redaktion »64'er«.

Verantwortlich: Für redaktionellen Teil: Michael M. Pauly.
Für Anzeigen: Peter Schrödel.

Vorstand: Carl-Franz von Quadt, Otmar Weber

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:

Markt & Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2,
8013 Haar bei München, Telefon 089/46 13-0, Telex 522052

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Chefredakteur: Michael M. Pauly (py)

Stellv. Chefredakteur: Michael Scharfenberger (sc)

Redakteure: aa = Albert Absmeier (130), ev = Volker Everts (278), kg = Karin Gößlinghoff (269), gk = Georg Klinge (130), rg = Christian Rogge (278)

Redaktionsassistent: Dagmar Zednik (237)

Fotografie: Janos Feister, Titelfoto: Alex Kempkens

Layout: Leo Eder (Litg.), Willi Gründl, Walter Höß, Cornelia Weber

Auslandsrepräsentation:

Schweiz: Markt & Technik Vertriebs AG, Alpenstrasse 14, CH-6300 Zug,
Tel. 042-2231 55/56, Telex: 862329 mut ch

USA: M & T Publishing, 2464 Embarcadero Way, Palo Alto, CA 94303; Tel.
001-4240 600; Telex 752351

Manuskripteneinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck und zur Vervielfältigung der Programmlistings auf Datenträger. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Herstellung: Klaus Buck (180), Leo Eder (181)

Anzeigenleitung: Peter Schrödel (156)

Anzeigenverkauf: Alfred Reeb (211)

Anzeigenverwaltung und Disposition: Michaela Hörl (171)

Anzeigenformate: 1/4-Seite ist 266 Millimeter hoch und 185 Millimeter breit (3 Spalten à 58 mm oder 4 Spalten à 43 Millimeter). Vollformat 297 x 210 Millimeter. Beilagen und Beihefter siehe Anzeigenpreisliste.

Anzeigenpreise: Es gilt die Anzeigenpreisliste Nr. 1 vom 1. März 1984.

Anzeigenrundpreise: 1/4 Seite sw: DM 7400,-. Farbzuschlag: erste und zweite Zusatzfarbe aus Europaskala je DM 1000,-. Vierfarbzuschlag DM 3000,-. Platzierung innerhalb der redaktionellen Beiträge: Mindestgröße 1/4-Seite

Anzeigen im Einkaufs-Magazin: Die ermäßigten Preise im Einkaufs-Magazin gelten nur innerhalb des geschlossenen Anzeigenteils, der ohne redaktionelle Beiträge ist. 1/4-Seite sw: DM 5400,-. Farbzuschlag: erste und zweite Zusatzfarbe aus Europaskala je DM 1000,-. Vierfarbzuschlag DM 3000,-. **Anzeigen in der Fundgrube:** Private Kleinanzeigen mit maximal 5 Zeilen Text DM 5,- je Anzeige. **Gewerbliche Kleinanzeigen:** DM 10,- je Zeile Text.

Auf alle Anzeigenpreise wird die gesetzliche MwSt. jeweils zugerechnet.

Vertriebsleitung, Werbung: Hans Hörl (114)

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebs GmbH, Pleningner Straße 100, 7000 Stuttgart 80 (Möhringen), Telefon (0711) 72004-0

Erscheinungsweise: 64'er, Magazin für Computerfans, erscheint monatlich, Mitte des Vormonats.

Bezugsmöglichkeiten: Leser-Service: Telefon 089/46 13-238. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen. Das Abonnement verlängert sich zu den dann jeweils gültigen Bedingungen um ein Jahr, wenn es nicht zwei Monate vor Ablauf schriftlich gekündigt wird.

Telefon-Durchwahl im Verlag:

Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089-46 13 und dann die Nummer, die in Klammern hinter dem jeweiligen Namen angegeben ist.

Alles über Modems und Akustikkoppler

In Amerika sind sie längst zum Alltag geworden; die Modems. Nun gibt es auch in Deutschland Akustikkoppler zu einem relativ günstigen Preis. Damit eröffnet sich auch für den Heimcomputer im Wohnzimmer die faszinierende Welt der Datenfernübertragung. Wir sagen Ihnen, was Sie alles brauchen (Treibersoftware, Schnittstellen), wie Sie damit umgehen und welche (legalen) Möglichkeiten es bereits gibt, auf Mailboxen zuzugreifen.



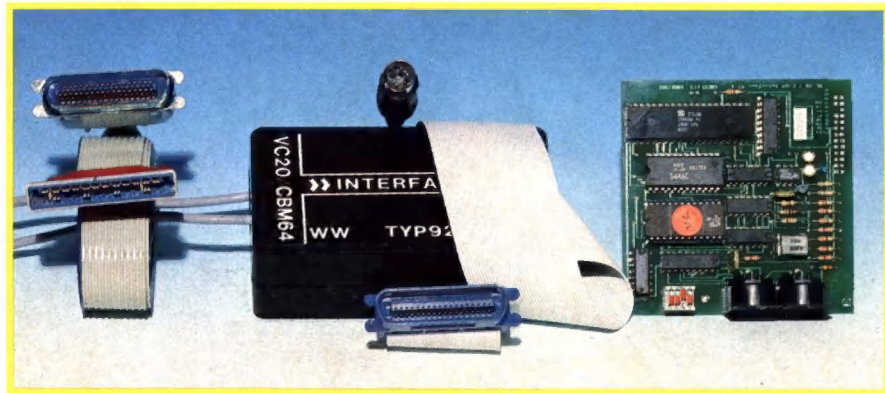
Programmiersprachen für den Commodore 64

Das Angebot an Programmiersprachen für den C 64 wird immer umfangreicher. Wir haben uns die beiden auf Mikrocomputern recht verbreiteten Sprachen Pascal und Forth einmal näher angesehen. Wie sieht die Konzeption dieser Sprachen aus, wie programmiert man damit und wie leistungsfähig sind sie? Neben diesen allgemeinen Betrachtungen stellen wir in mehreren Folgen vier Pascal-Compiler (Pascal 64, Abacus-Pascal, Kmmm-Pascal) und zwei Forth-Versionen (Power-Forth und 64 Forth) für den C 64 vor.

Adressenvergleich Commodore 64 — VC 20

Ein leidiges Thema: Wie kann man Programme von dem einen Computer auf den anderen umschreiben? Welche speziellen Systemeigenschaften sind zu beachten? Unser Adressenvergleich zwischen dem Commodore 64 und dem »kleinen Bruder« VC 20 bringt alle Informationen.

Centronics-Schnittstellen



Nicht jeder möchte einen Commodore-Drucker an seinen VC 20/C 64 anschließen. Allerdings besitzen andere Drucker oftmals nur eine Centronics-Schnittstelle. Wir haben für Sie einige auf dem deutschen Markt erhältliche Interfaces getestet. Unser Test gibt Ihnen Entscheidungshilfen, denn nicht jede Schnittstelle unterstützt alle Möglichkeiten Ihres Epson-Druckers. Sie können gespannt sein.

Magic Desk

Commodores »magischer Schreibtisch« ist das erste einer Reihe von Anwenderprogrammen, die auch dem völligen Laien das Arbeiten mit dem Computer ermöglichen sollen. Statt der Eingabe mehr oder weniger abstrakter Befehlsfolgen wird »Magic Desk« durch das »Antippen« von Bildern per Joystick gesteuert. Unser Testbericht zeigt Leistungsfähigkeit und Grenzen dieses Systems.

Relative Dateien

Die relative Datei bietet gegenüber der sequentiellen Datenspeicherung einige Vorteile. Bei ihr spielt der Arbeitsspeicher keine große Rolle. Entscheidend ist hier die Diskettenspeicherkapazität. Sie bestimmt die maximale Größe der relativen Datei. Somit kann man auch große Datenmengen selbst mit dem VC 20 und dem C 64 verwalten. Wir zeigen, wie sich jeder seine relativen Dateien programmieren kann.

CP/M-Software

Wie sieht die Praxis beim Einsatz von CP/M-Software auf dem C 64 aus? Lohnt sich das Übertragen von Programmen anderer CP/M-Computer und was kann man mit der Systemdiskette alles anfangen? Wie gut ist zum Beispiel Wordstar auf dem C 64?

Basic Bär

Das Programm, das Programme schreibt. Was dieser Programmgenerator leistet und wie komfortabel er sich handhaben läßt, haben wir getestet.

Listings

— Anwendung des Monats:

C 64 gießt Ihre Blumen

— Listing des Monats:

Der C 64 als Flipper

— Kurvendiskussion mit Hi-Res-Grafik und Hardcopy auf dem VC 1526

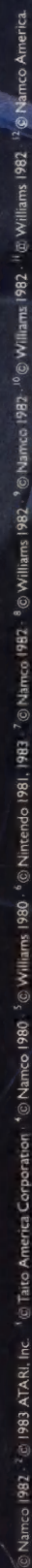
— Q-Bernd — fast besser als das Original

— Autostart mit der Floppy in Theorie und Praxis

— Programmverwaltung: Der eigene Programmkatalog

— außerdem natürlich wieder viele Tips und Tricks für den C 64 und den VC 20

© Namco 1982.² © 1983 ATARI, Inc.³ © Taito America Corporation.⁴ © Namco 1980.⁵ © Williams 1980.⁶ © Nintendo 1981, 1983.⁷ © Namco 1982.⁸ © Williams 1982.⁹ © Williams 1982.¹⁰ © Williams 1982.¹¹ © Williams 1982.¹² © Namco America.



ATARI[®]
© 1984 Atari, Inc. All rights reserved. Änderungen vorbehalten